

HP 13255

2648A OPERATING SYSTEM MICROCODE

Manual Part No. 13255-90010

PRINTED

APRIL-17-78

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied or reproduced without the prior written consent of Hewlett-Packard Company.

Copyright c 1978 by HEWLETT-PACKARD COMPANY

NOTE: This document is part of the 264XX DATA TERMINAL product series Technical Information Package (HP 13255).

CONTENTS

SECTION	CONTENTS	PAGE
1.0	INTRODUCTION	2
1.1	PURPOSE	
1.2	SCOPE	
1.3	RELATION TO 2645 FIRMWARE	
2.0	MEMORY ALLOCATION	2
2.1	ROM ADDRESS MAP	
2.2	DISPLAY MEMORY MAP	
2.3	FAST RAM MAP	
3.0	2645 FIRMWARE CHANGES	7
3.1	MAIN CODE CHANGES	
3.2	KEYBOARD CODE CHANGES	
3.3	DATACOM CODE CHANGES	
3.4	I/O CODE CHANGES	
4.0	GRAPHICS VARIABLES	14
4.1	FAST RAM	
4.2	DISPLAY RAM	
4.3	AVAILABLE MEMORY	
5.0	GRAPHICS ROUTINES	16
5.1	VECTOR DRAWING	
5.2	DISPLAY CONTROL	
5.3	GRAPHICS TEXT	
5.4	FLOATING POINT ROUTINES	
5.5	CHANGING THE GRAPHICS KEYPAD	
6.0	DELETING AUTO PLOT	29
6.1	ENTRY VECTORS	
6.2	RAM USED BY AUTO PLOT	
7.0	CONTROLLING THE GRAPHICS HARDWARE	34
7.1	HARDWARE OVERVIEW	
7.2	HARDWARE STATUS	
7.3	STROBES	
7.4	REGISTERS	
7.5	CLEARING OR SETTING THE SCREEN	
7.6	VECTOR GENERATION	
7.7	GRAPHICS CURSOR	
7.8	ZOOM	
7.9	SELF TEST	
7.10	READING THE IMAGE MEMORY	

1.0 INTRODUCTION

1.1 PURPOSE

This document describes the firmware implementation of the 2648A graphics terminal. Only selected topics will be covered, and that coverage will necessarily be brief. The two goals of this document are:

1. To give users entry points to routines that can be immediately useful, such as those for drawing vectors, turning zoom on and off, etc. The routines selected are those executed when graphics escape sequences and keystrokes are executed.
2. To describe the firmware/hardware interface sufficiently so that the implementation of the above routines can be understood and expanded upon.

1.2 SCOPE

It will be assumed that the user is familiar with the operation of the 2645A and its firmware, on which the 2648A is based. Only those features peculiar to graphics are discussed.

1.3 RELATION TO 2645A FIRMWARE.

The 2648A firmware is an extension of that developed for the 2645A. The bulk of the graphics additions consist of subroutines added to service the new escape sequences which control graphics functions. While virtually every 2645 module was changed, as outlined in Section 3, these changes are relatively minor. Virtually all of the information contained in the 2645A Operating System Microcode manual, part number 13255-90003, is also applicable to the 2648A.

2.0 MEMORY ALLOCATION

The 2648A's microprocessor can address 64K bytes of memory. In general, the range 0 to 48K contains the microcode (in ROM), 48K to 52K is RAM used for buffers, and 52K to 64K is RAM used for display memory. The range 36K to 36.5K contains fast access RAM. The range 32K to 36K is reserved for memory mapped I/O. (See figure 1.)

2.1 ROM ADDRESS MAP

The firmware is implemented as discrete modules. Each module can be separately assembled. A module references routines in other modules through entry vectors stored at the beginning of each module. The size and location of each module is as follows:

MODULE	SIZE	LOCATION
Main Code (essentially 2645 main code)	10K	0 - 10K
Graphics	18K	24K - 32K 38K - 48K
I/O	8K	10K - 18K
Keyboard	2K	18K - 20K
Datacom	2K-4K	20K - 24K
Alternate I/O	1.5K	36.5K - 38K

2.2 DISPLAY MEMORY MAP

Display memory resides from 48K to 64K. The topmost portion is used by the rom code for variable storage and buffers. The end of this variable section is set by the Main Code equate DSPLIM. As this must be on a 256 byte boundary, there are some unused memory locations. (See figure 2.)

2.3 FAST RAM MAP

Each Control Store PCA (02640-60192) contains 256 bytes of RAM. The access time for this memory is much faster (at least twice as fast) than for display memory. The 2648A contains two blocks (512 bytes) of this RAM. The microprocessors stack is stored in the first block, and grows downward towards the second (graphics) block. Under normal operation the stack does not reach the graphics fast RAM area. (See figure 3.)

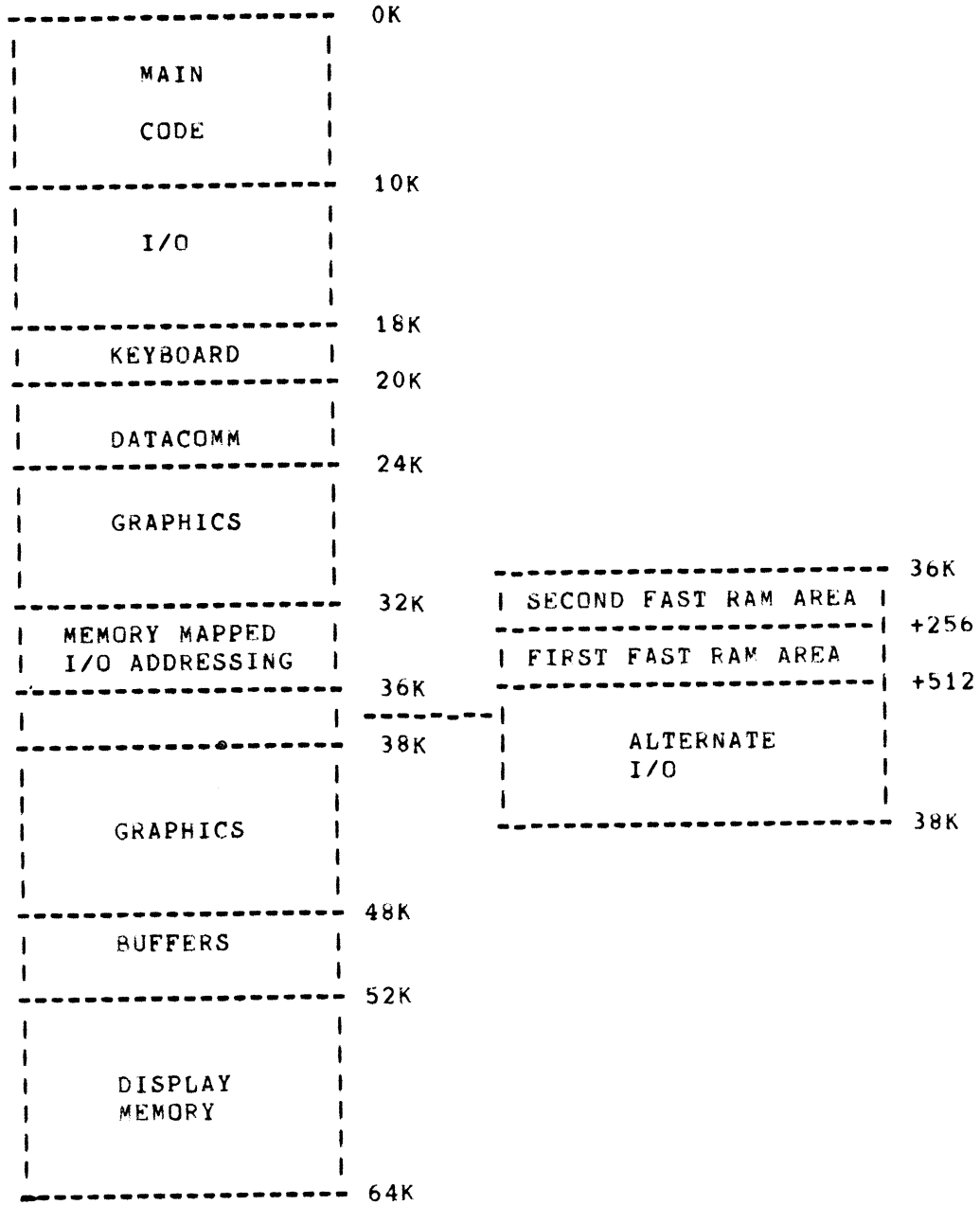


FIGURE 1

-----	177777B (OCTAL)	
COMMON VARIABLES	(48 bytes)	
-----	177720B	
MAIN CODE	(176 bytes)	
VARIABLES		
-----	177440B	
KEYBOARD VARIABLES	(32 bytes)	
-----	177400B	
DATA COMM	(128 bytes)	
VARIABLES		
-----	177200B	
I/O VARIABLES	(24 bytes)	
-----	177150B	
ALTERNATE I/O	(24 bytes)	
-----	177120B	
MESSAGE BUFFER	(80 bytes)	
-----	177000B	
DEVICE		
I/O	(512 bytes)	
BUFFERS		
-----	176000B	
GRAPHICS VARIABLES	(294 bytes)	
-----	175332B	
AUTO PLOT	(589 bytes)	
MENU		
-----	174215B	
UNUSED	(142 bytes)	
-----	173777B (DSPLIM)	
DISPLAY		
AREA		
////////////////////		
	(up to 12K bytes total)	
////////////////////		
-----	DSPBGN	

UNCHANGED FROM
2645 FIRMWARE

FIGURE 2



FIGURE 3

3.0 2645 FIRMWARE CHANGES

This section briefly describes the major changes made to 2645 modules to accomodate graphics. The description is limited to naming the affected routine and giving the purpose for the change.

3.1 MAIN CODE CHANGES

3.1.1 CHINT AND RANGE TABLES

The single most important main code change was to the range table mechanism to allow subroutine addresses greater than 32K. Consider the following example from the 2645 code:

```
*****  
; NORMAL CHARACTER SET ATTRIBUTES *  
*****  
RTABLE EQU $-3  
      DB 400,1770 ;ALPHANUMERICS  
      DW DSPCHR+B15 ;DISPLAYABLE CHARACTERS  
      DB 70,170 ;BELL,BS,HT,LF,VT,FF,CR,SO,SI  
      DW RTB010 ;USE INDEX TABLE  
      .  
      .  
      .  
;   
; <BELL> THROUGH <SHIFT IN>  
;  
RTB010 EQU $  
      DW ZBELL ;BELL-SOUND KEYBOARD BELL  
RTB020 EQU $  
      DW BCKSPC ;<BS> THROUGH <SHIFT IN>  
      DW HTAB ;BS - BACKSPACE CURSOR  
           ;HORIZONTAL TAB  
      .  
      .  
      .  
      .
```

Each range table entry consists of two lines. The first contains an upper and a lower bound. The second contains an address. If the MSB of the address is set, it indicates that the address is a subroutine address to be executed directly. The MSB is masked off to get the true subroutine address. If the MSB is not set, then the address is that of an index table. The subroutine address must be extracted from the table. When a character arrives at the terminal from datacom, keyboard, or tape, the routine CHINT determines what action should be taken by comparing the character to the upper and lower bounds in the current range table. When a match is found, the indicated subroutine is executed.

Note that no index table can have an address greater than 32K, or the MSB would be set, indicating a direct jump address. Also, no direct jump can be to an address greater than 32K, since the MSB is assumed to be a flag, and is masked off.

Since much of the graphics code is above 32K, these problems were solved as follows:

1. All index tables were placed below 32K. If a range table address entry has its MSB cleared, then it is taken to be an index table address.
2. Chint and the form of the range tables were modified to allow direct jump addresses greater than 32K. If a range table address has its MSB set, it is taken to be a jump address. However, after its MSB is masked out, another bit is OR'ed in. By setting this bit, an address in the full 0-64K range can be reached. The bit used is the MSB of the upper bound entry. Upper and lower bound entries use only 7 bits in the character comparison, as only ASCII characters are expected.

A typical graphics range table entry appears as follows:

```
*****  
; PLTTAB--USED IN VECTOR PLOTTING SEQUENCE *  
*****  
PLTTAB EQU $-3  
DB 40Q,PLTPRM/XDIV*XMUL+77B ;PARAMETER  
DW PLTPRM+B15  
DB 141Q,154Q ;SMALL A-L  
DW PINDX ;USE INDEX TABLE  
.  
.  
.  
PINDX EQU $  
DW PENUP ;A--RAISE PEN  
DW PENDN ;B--LOWER PEN  
.  
.  
.  
.
```

If the MSB of the address entry is not set, the address is used as an index table address. All index tables are below 32K, so the MSB will never be set.

If the MSB is set, the address is taken to be a subroutine address. The MSB is masked off and replaced by the MSB of the upper bound entry, and control is transferred to that location.

The operation PLTPRM/XDIV*XMUL shifts the MSB of the address (a 16 bit value) to the location of the MSB of the upper bound (an 8 bit value). If the routine PLTPRM was located above 32K, the bounds entry would be assembled as
DB 40Q,177Q. If below 32K, the bounds entry would be
DB 40Q,077Q.

To summarize,

1. Index tables are indicated if the MSB of the address is not set.
2. A direct jump address is indicated if the MSB of the address is set. The effective address is computed by replacing the MSB of the table address with the MSB of the upper bound entry.

3.1.2 SOFT KEY CHANGES

The carriage return key was made into a soft key. GETKEY returns a keycode of 357B for that key, instead of 15B. The soft key definition routine DFSFKY was expanded accordingly, as was the waitloop.

The routine EXSFKY was added to allow execution of soft keys through an escape sequence. This also involved changes to DFSFKY to recognize the triggering sequence.

3.1.3 GRAPHICS TEXT

Routines that add characters to the display, CHINT and DSPCHR, were modified to send characters to graphics if graphics text mode is on. Certain control codes and cursor positioning routines are also executed in graphics when graphics text mode is on. The relevant routines are CRRET, LNFEED, BCKSPC, and HTAB for the control codes, and CURPR, CURPL, CURPD, and CURPU for cursor positioning.

3.1.4 ALPHANUMERIC INHIBIT

Both the alphanumeric display and the alphanumeric cursor can be inhibited. The display is inhibited by inserting an 'END OF PAGE' (EOP) code as the first character of the display (the location DISPST). Routines that would change this location (TOPUPD, DSPMSG, SFKYON, and RSTDSP) now check graphics flags to determine the state of the display. The alpha display is re-enabled, and the graphics display inhibited, to display error messages and soft keys.

The alpha cursor is inhibited by directing the display hardware to put it off the screen. This is done by storing a cursor row greater than 23 in the location IDCCRW. Routines which change the cursor row address now go through the graphics routine ANCHK to disallow such stores if the cursor is inhibited.

3.1.5 CHANGES FOR AUTO PLOT

Changes for autoplot were made to direct incoming characters to the plot routine, and to allow proper interaction with the menu. The plot routine is called from CURADV (which is called whenever a character is put on the screen) as well as from certain control codes. CR and LF cause jumps to autoplot directly from the range tables. Changes for the menu were made in ESCEND, to restore the proper range table, and LOCLIN, to allow local only entry into the menu, and SFKYDS. To allow recording of the menu, GETDSP and INITDG were changed.

3.1.6 BLOCK TRANSFERS

Graphics status requests cause block transfers to be initiated. The routines DSPTCH and CLPLXF were changed to increase the number of possible pending transfers.

3.1.7 ECHO SUPPRESSION

In order to prevent status transfers from being displayed if they are echoed, GETDCM was modified to ignore datacomm input if an echo suppress flag is set. This mode is turned off (in the same routine) upon receipt of certain control codes.

3.1.8 VERTICAL RETRACE SCAN

The graphics cursor, the rubber band line, and zoom are updated once per frame. Consequently, the normal waitloop scans of datacom and keyboard were expanded to monitor a vertical retrace flag. When set, it indicates that one frame has elapsed.

3.2 KEYBOARD CODE CHANGES

3.2.1 RETSCN ROUTINE

The routine RETSCN was removed from the I/O code, due to lack of space there, and put into the keyboard ROM, which has extra space. Entry to the routine is made through entry vectors at the start of the keyboard module.

3.2.2 ASCII TABLES

The upper and lower case ASCII tables, UPRASC and LWRASC, were modified to replace the numeric pad with graphics functions, as well as to make the return key a soft key. The new values returned are as follows:

2645 KEY	2645 CODE	2648 CODE	2648 FUNCTION	SHIFTED 2648 CODE	SHIFTED 2648 FUNCTION
CR	15B	357B	SOFT RETURN	357B	SOFT RETURN
0	60B	245B	CURSOR FAST	222B	MENU
1	61B	213B	ZOOM IN	215B	CLEAR
2	62B	243B	CURSOR DOWN	226B	T ANG
3	63B	214B	ZOOM OUT	227B	T SZE
4	64B	244B	CURSOR LEFT	220B	DRAW
5	65B	207B	STOP	217B	A DSP
6	66B	242B	CURSOR RIGHT	221B	MOVE
7	67B	223B	AUTO PLOT	224B	AXES
8	70B	241B	CURSOR UP	211B	RB LN
9	71B	210B	G CURSOR	216B	G DSP
.	56B	212B	ZOOM	225B	TEXT

3.2.3 GRAPHICS CURSOR KEYS

The graphics cursor keys (codes 241B to 245B) are scanned separately to allow more than one to be depressed, for diagonal movement and two speed operation. In the keyboard routine GTKEY, when a key is depressed the graphics routine GKKEYS is called. When a key is released, the graphics routine RELGC is called. These routines test the key in question to see if it is a graphics cursor key, and if it is, bypass the normal GTKEY processing.

3.2.4 SOFT RETURN KEY

The carriage return key was made into a soft key and consequently returns a keycode of 357B instead of 15B.

3.2.5 TIMER INTERRUPT

The graphics timer interrupt routine is called from the keyboard routine KBMON.

3.2.6 REPEAT RATES

The initial delay and the repeat rates for graphics keys are set for longer than normal keys in GTKEY and SETRPT.

3.2.7 SETTING STRAPS P AND Q

A test for an escape sequence setting straps P or Q is made in the routine STJMPR.

3.3 DATA COMM CODE CHANGES

3.3.1 BUFFER ALLOCATION

If either strap P or Q is removed at power on, a very large data comm buffer (2048 bytes) is allocated, instead of the normal 96 byte buffer. The affected routines are DCINTR, GETDC, and INITDC.

3.3.2 ENQ/ACK HANDSHAKE

If in either scaled or unscaled Tek mode, ACK is not returned upon receipt of ENQ. The affected routine is GETDC.

3.3.3 RUBOUT CHARACTERS

If in either scaled or unscaled Tek mode, rubout characters (177B) are not stripped out of the input data. (DCINTR)

3.3.4 MULTIPOINT

The multipoint firmware is unchanged.

3.4 I/O CODE CHANGES

ALTERNATE I/O

Alternate I/O was moved from 24K to 36.5K. The new entry vectors are as follows.

LOCATION (OCTAL)	FUNCTION
111002B	Initialization Routine
111005B	Initialization Continuator
111010B	Interrupt Processor
111013B	Monitor Routine
111016B	Input Routine
111021B	Output Routine
111024B	Control Routine
111027B	Status Routine
111032B	Device Name Message

3.4.2 SOFT RETURN KEY

Routines which monitor the keyboard for a carriage return (GRNKEY, USRSKP, USREOF) now check for the soft return key keycode.

3.4.3 RETSCN ROUTINE

The I/O routine RETSCN was moved to the keyboard module due to lack of space.

3.4.4 TRANSPARENT READ

When reading buffer to display (BF2DSP) in Tek mode, the buffer is copied exactly as is. CR/LF's are not stripped out of the buffer, nor are they appended at the end of the record. Carriage return is used to signal the end of a Tek graphics sequence, and should not be automatically added by the terminal.

3.4.5 AUTO PLOT GET

In the display to buffer routine (DSP2BF) tests are made to see if the Autoplot menu is being displayed.

3.4.6 I/O BUFFER FLUSH

when record mode is turned off in RCRDGO, a test is made for the I/O buffer being partially filled. If it is, the data is recorded before terminating record mode.

3.4.7 PRINTER DRIVER

The RS232 printer driver now interprets a switch configuration of all closed, which used to mean send 56 nulls after each control code, as meaning send NO nulls (PTRCHR). Previously, at least one null was always sent. In addition, the Clear to Send line is no longer monitored (PRCHR2).

4.0 GRAPHICS VARIABLES

This section describes the use of selected variables used by the graphics code to store parameters and flags. A flag is a single bit which when set indicates that the terminal is a certain state or that a certain action is to be taken. For example, when zoom is to be turned on, the flag WANTZM is set in the variable GFLGS5.

4.1 FAST RAM VARIABLES

GFLGS1--Graphics Flags

MOVE = 1 => MOVE PEN WITHOUT DRAWING VECTOR
AVINHB = 1 => ALPHA VIDEO INHIBITED
ACINHB = 1 => ALPHA CURSOR INHIBITED

GFLGS3--Graphics Flags

WANTGC = 1 => USER WANTS GRAPHICS CURSOR TURNED ON
WANTRB = 1 => USER WANTS RUBBER BAND LINE ON

GFLGS5--Graphics Flags

WANTZM = 1 => USER WANTS ZOOM TURNED ON

GFLGS6--Graphics Flags

GTEXT = 1 => GRAPHICS TEXT MODE TURNED ON
SLANT = 1 => SLANT GRAPHICS TEXT

TKFLGS--Graphics Flags

UNSCLD = 1 => UNSCALED TEK MODE ON
SCLD = 1 => SCALED TEK MODE ON
SUPCHR = 1 => ECHO SUPPRESS TURNED ON

XCURR, XCURR+1, YCURR, YCURR+1--16 bit signed value of the current pen position.

CURCD--Flags indicating whether the current pen position is within the clipping boundaries or not.

LTXMIN = 1 => X COORD IS LESS THAN MINIMUM ALLOWABLE
GTXMAX = 1 => X COORD IS GREATER THAN MAXIMUM ALLOWABLE
LTYMIN = 1 => Y COORD IS LESS THAN MINIMUM ALLOWABLE
GTYMAX = 1 => Y COORD IS GREATER THAN MAXIMUM ALLOWABLE

XMIN, XMIN+1--Minimum X clipping value (see Section 5.1.2).

XMAX, XMAX+1--Maximum X clipping value.

YMIN, YMIN+1--Minimum Y clipping value.

YMAX, YMAX+1--Maximum Y clipping value.

CURMOD--Current vector drawing mode (see Section 5.1.4).

CURPAT--Current dot-dash pattern.

SCALE--Current pattern scale factor.

CURGCX, CURGCX+1, CURGCY, CURGCY+1--Current graphics cursor position.

PRMBUF--16 byte buffer for storing incoming graphics parameters.

4.2 DISPLAY RAM

MAG--Current zoom magnification (0-15).

ZX, ZX+1, ZY, ZY+1--Current coordinates of center of zoomed area.

NUMBUF--132 character buffer used for graphics text and messages (also equated as LBLBUF).

TXMAG--Current graphics text size (0-7).
TXANG--Current graphics text orientation (0-3).
TXORG--Current graphics text origin (0-8).

APFLGS--Autoplot flags

APIP = 1 => AUTO PLOT CHARACTER SCAN IS TURNED ON

APFLG2--Autoplot flags

APMUON = 1 => AUTO PLOT MENU IS TURNED ON

4.3 AVAILABLE MEMORY

The following memory locations are available for use:

1. Fast RAM between 110340B and 110377B (32 bytes).
2. Fast RAM between 110100B and 110120B (16 bytes).
3. Display RAM between 173777B and 174215B (142 bytes).
4. Display RAM between 175711B and 175722B (10 bytes).

5.0 GRAPHICS ROUTINES

This section lists some of the important routines used to control graphics functions. A brief description of the purpose of the routine, the symbolic label used to address it, and any registers used for passing parameters are given. Unless specified otherwise, registers not listed are assumed to be destroyed.

Some of the routines exit through GEXIT. This routine tests the escape sequence character used to invoke the character for upper or lower case. If upper case, the escape sequence is terminated through ESCEND. To prevent the range tables from being changed by ESCEND, if a routine that exits through GEXIT is called, a lower case character should be stored in the location ZCHAR (177610B).

5.1.1 DRAWING VECTORS

VECTOR--DRAW VECTOR OR CHANGE CURRENT PEN POSITION WITHOUT
DRAWING

ENTRY XNEW, XNEW+1 = NEW X COORDINATE
YNEW, YNEW+1 = NEW Y COORDINATE
XCURR, XCURR+1 = CURRENT PEN X COORDINATE
YCURR, YCURR+1 = CURRENT PEN Y COORDINATE

GFLGS(MOVE) = 0 => DRAW VECTOR FROM XCURR, YCURR TO
XNEW, YNEW

GFLGS(MOVE) = 1 => MOVE PEN TO XNEW, YNEW WITHOUT
DRAWING

CURPAT, SCALE = CURRENT DOT-DASH PATTERN AND SCALE

CURMOD = CURRENT DRAWING MODE

The X and Y values are 16 bit signed (2's complement) values. They can be outside the range of the visible screen (0 to 719 for X, 0 to 359 for Y), as the routine tests to see if they are in bounds or not. If not, only the portion of the vector that is on screen, if any, is drawn. This process is called clipping. It is necessary to use this routine to move the pen without drawing so that it can be determined whether the new coordinates are on screen or not, and set the flags in CURCD accordingly.

The dot-dash pattern and drawing mode used when generating the vector can be changed by the routines described later in this section.

5.1.2 CHANGING THE CLIP LIMITS

VECTOR only draws the portion of the vector within a clipping region. It is possible to change this region so that an area smaller than the normal 0-719 X, 0-359 Y is used. This is done when autoplot is running. The new limits are stored as the 2's complement of the desired 16 bit value in the following locations:

XMIN,XMIN+1 - Lower left X coordinate of area (normally 0).
YMIN,YMIN+1 - Lower left Y coordinate of area (normally 0).
XMAX,XMAX+1 - Upper right X coordinate of area (normally -719).
YMAX,YMAX+1 - Upper right Y coordinate of area (normally -359).

After the clip limits have been changed, the routine HRD2 should be called to update the out-of-bounds flags, as the current pen position may now be outside (or inside) the clipping region. The normal clipping limits can be set by calling HRD1.

A discussion of clipping can be found in the book "PRINCIPLES OF INTERACTIVE GRAPHICS", by Newman and Sproull. The algorithm used is described in the paper "A CLIPPING DIVIDER", AFIPS 1968 Fall Joint Computer Conference, Vol. 33, Part 1, pages 765-775.

5.1.3 CHANGING LINE TYPE

SETLN1--CHANGE LINE TYPE

ENTRY A = NEW LINE TYPE
0 = SOLID
1 = USER DEFINED DOT-DASH
2 = USER DEFINED AREA SHADING
3-9 = PREDEFINED DOT-DASH
10 = POINT PLOT

This routine changes the pattern used when drawing vectors. The value of the predefined patterns is defined by the table LINETB. The simplest way to generate user-defined patterns is by sending an image of the appropriate escape sequence through CHINT.

5.1.4 CHANGING DRAWING MODE

SETMD1--SET DRAWING MODE

ENTRY A = NEW DRAWING MODE
0 = DO NOTHING
1 = CLEAR
2 = SET
3 = COMPLEMENT
4 = JAM PATTERN

This routine changes the way the image memory is modified when vectors are drawn. Do nothing mode leaves the image memory unchanged, essentially a no-op. It is useful only when reading the image memory. Set, clear and complement perform the appropriate operation only if the pattern bit which is to be drawn is a '1'. That is, if a solid vector is being drawn (all pattern bits on) then each bit in the vector would be set, cleared, or complemented. If the pattern had only every other bit on (a dotted line) then only every other bit of the vector would cause the image memory to be modified. When set, clear, or complement mode is selected, a '0' in the pattern does not modify the image memory at all. Jam pattern mode copies the pattern directly into the image memory. A '1' in the pattern causes the memory bit to be set, and a '0' causes the memory bit to be cleared.

5.2 DISPLAY CONTROL

5.2.1 CLEARING/SETTING THE IMAGE MEMORY

GCLR1--CLEAR THE IMAGE MEMORY
GSET1--SET THE IMAGE MEMORY

ENTRY DON'T CARE

These routines write 0's and 1's, respectively, into the image memory. They operate as the image memory is being read to display the graphics image. Consequently, it takes only one frame time (approximately 16 milliseconds) to write the entire memory. When the display is zoomed, only a portion of the memory is read, so that only that part being displayed (approximately) is written. More than is being displayed may be cleared while zoomed since the memory can only be read 16 bits at a time. If the zoom location starts or ends in the middle of a 16 bit word, the entire word will be written.

5.2.2 GRAPHICS VIDEO ON/OFF

GVON1--TURN GRAPHICS VIDEO ON
GVOFF1--TURN GRAPHICS VIDEO OFF

ENTRY DON'T CARE

These routines enable and disable the graphics video signal generated by the graphics hardware. A hardware gate prevents the graphics signal from reaching the terminal's display subsystem. Consequently, the image memory is unchanged.

With the graphics video off, the graphics hardware does not have to read the image memory to generate the video signal, which it normally spends 75% of its time doing, and can instead spend more time drawing vectors. With the video off, the vector generator can draw vectors approximately 4 times faster than with it on. This difference is not normally noticeable for single vectors, since the firmware overhead is the limiting factor. Area filling operations make better use of this feature.

5.2.3 ALPHANUMERIC VIDEO ON/OFF

ANVON1--TURN ALPHANUMERIC VIDEO ON
ANVOF1--TURN ALPHANUMERIC VIDEO OFF

ENTRY DON'T CARE

These routines enable and inhibit the alpha video without erasing the alpha memory. The display is inhibited by inserting an 'END OF PAGE' code as the first character of the display. The hardware fills the rest of the display with blanks.

5.2.4 GRAPHICS CURSOR ON/OFF

TGCON1--TURN GRAPHIC CURSOR ON
TGCOF1--TURN GRAPHIC CURSOR OFF

ENTRY DON'T CARE

The rubber band line is also turned off when the cursor is turned off. If the cursor is turned on when graphics text mode is on, it is moved to where the next character would be drawn, so that it can act as a text cursor.

5.2.5 ALPHA CURSOR ON/OFF

ACON1--TURN ALPHA CURSOR ON
ACOFF--TURN ALPHA CURSOR OFF

ENTRY DONT CARE

EXIT ACOFF EXITS THROUGH GEXIT

The alpha cursor is inhibited by telling the hardware to put it off screen. Flags prevent routines which update the cursor position from sending its true location to the hardware. Since the cursor is updated after every character is put on the screen, these flags must be examined for every character. Consequently, the terminal cannot put characters on the screen as fast as the 2645.

5.2.6 ZOOM ON/OFF

ZON1--TURN ZOOM ON
ZOFF1--TURN ZOOM OFF

ENTRY DON'T CARE

When zoom is turned on, the region centered about the graphics cursor position is zoomed by the current zoom size.

5.2.7 SET ZOOM SIZE

NWSIZE--SET NEW ZOOM SIZE

ENTRY A = ZOOM SIZE (0-15)

The zoom size can be set while zoom is either on or off.

5.2.8 SETTING CURSOR OR ZOOM POSITION

GCP1--SET GRAPHICS CURSOR POSITION
ZPOS1--SET ZOOM POSITION

ENTRY PRMBUF, PRMBUF+1 = 16 BIT X COORDINATE
PRMBUF+2, PRMBUF+3 = 16 BIT Y COORDINATE

With either routine, the graphics cursor is moved to the specified point. If a coordinate is out of bounds, the value will be set to the maximum or minimum allowable value. The zoom position specified will be the center of the zoom area. Changing the cursor position while zoomed will not change the zoom area until the cursor reaches the edge of the area. Then the cursor 'drags' the zoom area, so that the cursor would be at the edge of the region, not the middle.

5.2.9 RUBBER BAND LINE ON/OFF

TRBON1--TURN RUBBER BAND LINE ON
TRBOF1--TURN RUBBER BAND LINE OFF

ENTRY DON'T CARE

The graphics cursor is also turned on when the rubber band line is enabled.

5.3 GRAPHICS TEXT

5.3.1 GRAPHICS TEXT MODE ON/OFF

GTXON1--TURN GRAPHICS TEXT MODE ON
GTXOF1--TURN GRAPHICS TEXT MODE OFF

ENTRY DON'T CARE

When graphics text mode is on all displayable alphanumeric characters are drawn in the graphics memory using the current size, orientation, slant, and origin. When the origin is set to left justify, each character is drawn at the current pen position, which is updated after each character. If right justify or center is selected, an entire line of characters is buffered until CR or LF is received, at which point the line is justified or centered about the current pen position. The pen position is updated by the CR and LF, so that succeeding lines will be justified about the proper point.

If the graphics cursor is on when graphics text mode is turned on, characters are drawn at the cursor position, the assumption being that the cursor is being used as a text cursor.

5.3.2 SET GRAPHICS TEXT SIZE

TXSIZ1--SET TEXT SIZE

ENTRY A = SIZE (0-7)

The size can be set while graphics text is on or off.

5.3.3 SET GRAPHICS TEXT ORIENTATION

ANGLE--SET TEXT ORIENTATION

ENTRY A = ANGLE (0-3)

This routine changes the direction in which characters are drawn. The parameter values have the following effect:

0 => Normal upright characters

1 => Rotate characters 90 degrees counter-clockwise

2 => Rotate characters 180 degrees counter-clockwise (upside down)

3 => Rotate characters 270 degrees counter-clockwise.

5.3.4 GRAPHICS TEXT SLANT ON/OFF

SLNTON--TURN SLANT ON

SLNTOF--TURN SLANT OFF

ENTRY DON'T CARE

EXIT BOTH EXIT THROUGH GEXIT

5.3.5 SET GRAPHICS TEXT ORIGIN

LORG1--SET ORIGIN

ENTRY A = ORIGIN (0-8)

The text origin is a single value which determines if a string of graphics text is to be left justified, right justified, or centered, using either the bottom, middle, or top of the character cell as the baseline. The relation between the parameter value and resulting string is as follows:

	2		5		8
	aaaaa	aaaaa	aaaaa	aaaaa	aaaaa
	a a a a a	a a a a a	a a a a a	a a a a a	a a a a a
1	aaaaa	aaaaa	a 4	a a a a a	7
	a a a a a	a a a a a	a a a a a	a a a a a	a a a a a
	a a a a a	a a a a a	a a a a a	a a a a a	a a a a a
0		3		6	

The normal setting is 0, left justified at the bottom of the character cell. 2, 5, and 8 imply putting the string below the specified point. 3, 4, and 5 cause centering, while 6, 7, and 8 cause right justification. When centering or left justifying, the characters are buffered so that the number of characters to be drawn, and from that the starting point of the string, can be determined.

This routine works as expected when the orientation is changed from normal upright characters.

5.3.6 PRINT CHARACTER BUFFER

SNDBUF--PRINT CONTENTS OF BUFFER IN GRAPHICS

ENTRY HL = POINTER TO FIRST CHARACTER OF BUFFER
A = NUMBER OF CHARACTERS

EXIT XCURR, YCURR UPDATED

The buffer of characters is drawn in the graphics memory using the current size, orientation, slant, and origin. Graphics text mode need not be turned on. The first character is drawn at the current pen position, which is updated after every character. Control codes in the buffer are ignored.

5.3.7 CHARACTER IMAGES

Graphics characters are drawn as a series of adjacent vectors, each with a particular dot-dash pattern. These patterns are stored in a table called CHRTAB. Each character has 10 pattern bytes associated with it. The basic character cell is 7 bits wide by 10 bits high. Only 7 of the 8 pattern bits are used. Larger characters are drawn by stretching the basic cell in both directions. To stretch it horizontally (assuming upright characters) the vector length and the pattern scale are multiplied by the desired size. The cell is stretched vertically by repeating the same pattern the proper number of times. A 4 X character, for example, is drawn with vectors 28 bits wide instead of 7, with a pattern prescale of 4 instead of 1. Each vector is repeated 4 times, so that the cell is now 28 X 40 instead of 7 X 10.

By changing the patterns in CHRTAB, the graphics character set can be redefined. The 10 pattern bytes are stored in 'top-down' order. The pattern definition includes blank dots at the top, bottom and sides of the character used for inter-character spacing. For example, the character capital A is defined as follows:

CODE	RESULTING CHARACTER
DB 000Q,070Q,104Q,104Q,104Q 000
DB 174Q,104Q,104Q,000Q,000Q	..@@@.. 070
	.@...@. 104
	.@...@. 104
	.@...@. 104
	..@@@.. 174
	.@...@. 104
	.@...@. 104
 000
 000

5.4 FLOATING POINT ROUTINES

The floating point routines used by autoplot are those available through the Intel users library, reference numbers BC1 and BC2. The following description is based largely on the Intel documentation. with the exception of INP, the routines are unchanged.

BC1 contains the floating point arithmetic routines only. BC2 contains the BCD and fixed point to floating point, and vice versa, conversion routines. The floating point representation uses 4 consecutive bytes to store a single value. 24 bits are used as the binary representation of the number, while 8 bits are used for the exponent. Note that the binary representation will cause round off problems. The largest number that can be represented is approximately 3.6×10^{38} . The smallest is approximately 2.7×10^{-39} . The floating point operations available are:

1. LOD--load a value from memory into the floating point accumulator.
ENTRY HL = pointer to 4 byte value to be loaded.
2. STR--store into memory the value in the floating point accumulator.
ENTRY HL = pointer to 4 byte store area.
3. ADD--add a specified value to the floating point accumulator.
ENTRY HL = pointer to 4 byte value to be added.
EXIT Floating point accumulator contains result,
processor flags set as appropriate (Z, NZ, M, P)
4. SUB--subtract a specified value from the F.P.A.
ENTRY HL = pointer to 4 byte value to be subtracted.
EXIT same as ADD
5. MUL--multiply F.P.A. by specified value.
ENTRY HL = pointer to 4 byte value to be multiplied.
EXIT same as ADD
6. DIV--divide F.P.A. by specified value
ENTRY HL = pointer to divisor
EXIT same as ADD
7. TST--set the processor flags to indicate the state of the F.P.A.
EXIT P, M, Z, NZ set as appropriate
8. CHS--change the sign of the F.P.A.
9. ABS--get the absolute value of the F.P.A.
10. INIT--an initializing routine that must be called (once only) before using DIV or MUL.

The conversion routines accept either strings or a fixed point format for conversion to floating point. The fixed point format uses 4 bytes to represent a 32 bit, signed (2's complement) number. The position of the binary point within the number is given by a binary scaling factor. This scaling factor is only used by the formatting routines. A value of 0 indicates that the binary point is immediately to the left of the most significant bit. A value of 32 indicates that the binary point is immediately to the right of the least significant bit. The scaling factor can have a value from -128 (200B) to +127 (177B).

Character strings for input consist of the ASCII representation for each character. This is the only change to the original Intel package, which used a different format for input strings. The following characters are valid in an input string:

Digits (0-9)	60B - 71B
Space	40B
+	53B
-	55B
.	56B
E	125B

Character strings on input may not cross a 256 byte boundary. An input string is terminated by the first character that departs from the specified format.

The output routine generates 2 possible formats, each 13 characters long. The format used depends on the magnitude of the value. Zero and values between .1000000 and 9999999. are represented by a space or minus sign, seven decimal digits, an appropriately positioned decimal point, and 4 spaces.

Magnitudes out of the above range are represented by a space or minus sign, a value between 1.000000 and 9.999999, a capital E, and a signed, two digit power of 10.

The output characters are not ASCII, but can be converted to ASCII by adding 60B to each.

CHAR	REPRESENTATION ON OUTPUT
Digits (0-9)	0B - 11B
Space	360B
+	373B
-	375B
.	376B
E	025B

Examples of input and output strings are as follows:

INPUT	OUTPUT
3.141593	3.141593
-.00000000000001	-1.000000E-13
+1.6E5	1600000.0
1.6	1.600000
123456789	1.234568E+08

The conversion routines are as follows:

1. FLT--convert from fixed point to floating point.
ENTRY A,B,C,D contain 32 bit signed fixed point value
(A = most significant byte, D = least significant)
E = binary scaling factor
EXIT Floating point accumulator contain floating point representation.
2. FIX--convert from floating point to fixed point.
ENTRY E = binary scaling factor
Floating point accumulator contains value
EXIT A,B,C,D contain fixed point value
(A = most significant byte, D = least significant)
3. INP--convert from ASCII string to floating point
ENTRY HL = pointer to start of string
(string is in true ASCII representation, and cannot cross a 256 byte boundary)
EXIT Floating point accumulator contains value
4. OU--convert from floating point to string
ENTRY HL = pointer to 13 byte buffer (cannot cross 256 byte boundary)
Floating point accumulator contains value to be converted
EXIT BCD representation of F.P.A. stored in buffer (must add 60B to each character to convert to ASCII)

5.5 CHANGING THE GRAPHICS KEYPAD

The graphics keypad returns key codes in the range 207B to 227B. The graphics routine K&FUNC uses the keycode as the index to a table, and branches to the address stored for key code it is called with. By changing the address stored in the table, it is very easy to redefine the graphics keys to perform any function desired. Each key returns two key codes, depending on whether the shift key is held down or not. The codes returned by each key are as follows:

Other routines could be substituted for these dummy routines, to use the existing hooks for alternate functions. The following section explains the purpose of each autoplot function called from an entry vector.

6.1 ENTRY VECTORS

6.1.1 TURN AUTO PLOT MENU OFF

APMUOF-- TURN THE AUTO PLOT MENU OFF

ENTRY DCN'T CARE

This routine is called to insure that the autoplot menu is off. It calls ESCEND, consequently the current range table will be reset.

6.1.2 APSCAN--SCAN INPUT DATA FOR AUTO PLOT

ENTRY DCHAR CONTAINS INPUT CHARACTER

EXIT ALL REGISTERS DESTROYED

This routine is called from the cursor advance routine CURADV in the main code whenever a character is put on the screen and autoplot is on. Numerical values are built up from single characters, and when numbers in the proper data columns are complete, a data point is plotted. Only displayable characters arrive at this routine.

MUCHK--SEE IF AUTO PLOT MENU IS UP

ENTRY DON'T CARE

EXIT Z => MENU NOT UP
NZ => MENU ON SCREEN
A DESTROYED

This routine is called to see if the autoplot menu is being displayed. Examples of main code routines that use it are ESCEND, to determine which range table to use, and GETDSP, to see if autoplot menu is to be read from the display memory.

INSFIX--COUNT NUMBER OF CHARACTERS INSERTED

ENTRY DON'T CARE

EXIT ALL REGISTERS SAVED
LOCATION 'INSERT' UPDATED

This routine is called only from the main code character display routine DISPLA, and is used by autoplot to keep track of the number of display enhancement codes inserted in a line. Autoplot uses this count to properly highlight numerical values in inverse video as the data values are scanned.

APCHK--AUTO PLOT KEYBOARD INPUT

ENTRY C = KEYBOARD CHARACTER

EXIT BC SAVED, ALL OTHERS DESTROYED

This routine is called from LOCLIN (main code) when autoplot is on and a keyboard entry is made. If any data point has have been plotted, autoplot is turned off. If no points have been plotted, the character is to be ignored by autoplot. This is done by incrementing an 'ignore' count, so that the character will not be processed by the autoplot scan routine when it is added to the display.

APCR--PROCESS CARRIAGE RETURN WHILE AUTO PLOT ON

ENTRY DON'T CARE

EXIT ALL REGISTERS DESTROYED
EXITS THROUGH MAIN CODE ROUTINE 'CRRET'

This routine is called directly from the range tables, and causes a normal carriage return to be executed when it is finished. Autoplot uses a carriage return to terminate any numerical value being built.

MUTB--ADDRESS OF MENU RANGE TABLE

The address of the autoplot menu's range table is used by the main code routine ESCEND. When an escape sequence terminates, ESCEND loads the appropriate range table depending on whether the normal display, the soft key display, or the autoplot menu is up.

GGTEST--TEST FOR GRAPHICS DATA GET

ENTRY DON'T CARE

EXIT A DESTROYED
NZ => GET GRAPHICS DATA
Z => GET ALPHA DATA

This is called by the main code routines GETDSP and INITDG (also some I/O routines) to determine which display data is to be used when reading the display memory. If the autoplot menu is up, the flags indicate that a graphics 'get' should be performed instead of the normal operation.

GGINIT--INITIALIZE FOR GRAPHICS GET

ENTRY DON'T CARE

EXIT Z => GRAPHICS DATA AVAILABLE
NZ => NO GRAPHICS DATA

This routine is called from INITDG if the data from the autoplot menu is to be read.

GRGET--GET GRAPHICS DATA

ENTRY DON'T CARE

EXIT NC => CHAR AVAILABLE
A => CHAR

CY => NO CHARACTER
M => END OF DISPLAY
P,NZ => END OF LINE

This routine returns the data stored in the autoplot menu one character at a time. Appropriate escape sequences for each menu field are generated. This routine is called from GETDSP, whenever the menu is recorded or copied to a printer, or when the Enter key is pressed while the menu is up.

HOME--HOME THE AUTO PLOT MENU CURSOR

ENTRY DON'T CARE

This routine puts the cursor in the first column of the first menu field.

APLF--PROCESS LINEFEED WHILE IN AUTO PLOT

ENTRY DON'T CARE

EXIT EXITS THROUGH MAIN CODE ROUTINE 'LNFEED'

Similar to APCR, this routine is called directly from the main code range tables. It is used to update the 'SKIP LINES' count. It causes a normal line feed to be executed.

6.2 RAM USED BY AUTO PLOT

6.2.1 FAST RAM

When autoplot is deleted, 64 bytes of fast ram become available, from 110000B to 110100B. This ram is used only by the floating point routines.

6.2.2 DISPLAY MEMORY

When autoplot is deleted, 768 bytes of display memory is freed. As seen from the address map in Figure 2, the display upper limit is normally set to 173777B. With the autoplot menu and variable store deleted, it is possible to raise the display limit (DSPLIM) to 175377B. This can only be done in the main code. If the DSPLIM equate is not changed, the free space will be unavailable to the display, but can be used for other variable storage.

6.2.3 VARIABLES USED BY OTHER MODULES

The main code reads the autoplot variable APFLGS to determine if autoplot is on or not. Using that location for something else will not cause problems (it is read only) as long as there are dummy routines for the autoplot entry vectors (which may be inadvertently called).

Fast ram variables must not be moved around if the floating point locations are deleted, since all modules reference variables in this area.

6.3 GRAPHICS KEYPAD

Three keys on the graphics keypad are used for autoplot functions. With autoplot deleted, these keys can be used to control other functions merely by changing the jump addresses associated with the following key codes.

KEY LABEL	CODE	FUNCTION
AUTO PLOT MENU	222B	TOGGLE AUTO PLOT MENU DISPLAY
AUTO PLOT	223B	START AUTO PLOT SCAN
AXES	224B	DRAW AUTO PLOT AXES

7.0 CONTROLLING THE GRAPHICS HARDWARE

This section describes how the microprocessor controls the graphics hardware. Further information on how the hardware works can be obtained from manual part number 13255-91125, Graphics M-Controller Module, and manual part number 13255-91126, Graphics Display Module.

7.1 HARDWARE OVERVIEW

The primary functions of the graphics hardware are as follows:

1. Reading the image memory in sync with the alphanumeric display subsystem (in either normal or zoom mode) to generate the graphics display.
2. Writing vectors into the image memory.
3. Clearing or setting the image memory.
4. Drawing the graphics cursor.
5. Performing self test.

7.1.1 IMAGE MEMORY ORGANIZATION

The graphics image memory contains one bit for every point on the 720 by 360 display. If this memory were organized as a two-dimensional X,Y array, it would require 10 bits (X) by 9 bits (Y), or 2^{19} bits to store the image. By assigning each image bit a number, it is possible to store the image as a one dimensional linear list 259,200 bits ($720 * 360$) long. A memory size of 2^{18} (262,144) bits is then sufficient, reducing the memory requirement by half.

This linear list is organized as 16,200 16 bit words. Each of the 16 16K ram chips contributes one bit to each word. Points adjacent on the screen are not necessarily adjacent in the memory. There are 8 possible memory displacements between adjacent screen points. Successive memory addresses correspond to screen dots along a horizontal line. Consequently, a complete scan line (720 dots) can be displayed by reading 45 contiguous words from the memory. A dot directly above another on the screen will be offset by 720 bits, one scan line, in the memory. Note that moving upward on the screen corresponds to a negative displacement. Since the raster sweeps top to bottom, the raster origin is taken to be the upper left hand corner of the screen, with increasing Y pointing downward. An X,Y coordinate is converted to a bit address by the relation:

$$\text{Bit Address} = (359-Y)*720 + X$$

The Y value is subtracted from 359 to compensate for the shifted origin.

7.1.2 VECTOR GENERATION

Vectors are generated by computing the memory addresses of the points on the screen which most closely approximate the line between specified endpoints. An iterative algorithm is used, where the memory address for a given point is computed by adding a memory displacement to the address of the previous point. For a vector in a given octant, there are only two possible displacements to choose from. The sign of a discriminant value determines which of the two to use at each point. After the initial values have been computed, the algorithm uses only addition and subtraction.

The initial values for the algorithm are computed by the microprocessor. These values include the initial starting point, converted from X,Y coordinates to an 18 bit memory address, the two memory displacements, the initial discriminant value, two discriminant increments, and the number of dots to be drawn. These values are transferred to registers on the graphics controller, which then executes the iterative algorithm.

7.1.3 VECTOR ALGORITHM

The description of the algorithm assumes a vector between the points (XSTART,YSTART) and (XFINISH,YFINISH) with absolute slope less than 45 degrees. For vectors of absolute slope greater than 45 degrees, Delta X and Delta Y should be swapped when computing D, D1, D2, and DC.

STEP 1. Compute the initial parameters, and transfer to the graphics controller.

Delta X = XFINISH - XSTART.
Delta Y = YFINISH - YSTART.
Initial Memory Address MA = 720*(359-YSTART)+XSTART.
Look up the memory displacements M1, M2 in a table using the octant (determined by Delta X and Delta Y) as a key.
Initial Discriminant D = -|Delta X| + 2|Delta Y|
Discriminant increment D1 = 2|Delta Y|
Discriminant increment D2 = 2|Delta Y| - 2|Delta X|
Dot Count DC = -(|Delta X| + 1)

STEP 2. Write the bit at Memory Address MA.

STEP 3. Set DC = DC + 1

If the Dot Count is 0, then stop, the vector is finished. (Note that the Dot Count is a negative value which is incremented by the hardware, NOT a positive value which is decremented.)

STEP 4. If the Discriminant D is negative,
set $D = D + D1$ (update the Discriminant)
set $MA = MA + M1$ (update the Memory Address)
GOTO Step 2

If the Discriminant D is positive,
set $D = D + D2$ (update the Discriminant)
set $MA = MA + M2$ (update the Memory Address)
GOTO Step 2

Further information on the derivation of the vector generation algorithm can be found in these sources:

1. "Algorithm For Computer Control of a Digital Plotter", J.E. Bresenham, IBM Systems Journal, Vol. 4, No. 1, 1965, 25-30.
2. "A Scan Conversion Algorithm With Reduced Storage Requirements", B.W. Jordan and R.C. Barrett, CACM, Vol.16, No. 11, Nov 1973, 681-682.
3. Hewlett-Packard Journal, Jan, 1978.

7.1.4 ARCHITECTURE

The hardware which implements the above algorithm is manipulated by the microprocessor through the use of 4 8-bit wide registers, 2 strobes, and 16 12-bit wide buffer locations. The overall method for controlling vector generation, cursor generation, zoom, and self test is to load the proper parameters into the buffer, then set flags in one of the registers to indicate what the buffer values are for. All register and buffer locations are loaded through normal 8080 memory reference instructions. A section of the memory address space is reserved for I/O addressing.

Since buffer locations are 12 bits wide, 2 separate 8 bit stores must be made to transfer all 12 bits. The address for loading the 4 upper bits is always one greater than that for the lower 8 bits. Consequently, the "Store H and L Direct" instruction can be used to send the L register into the 8 lower bits, and the 4 LSB of the H register into the 4 upper bits of the buffer location. Certain buffer locations use only one bit as a flag. The flag is set by storing 10B into the upper 4 bits of the buffer location. This corresponds to the MSB of the 12 bit value. The flag is cleared by storing 0. The lower 8 bits are ignored.

Memory addresses and increments, as explained in Section 7.1.1 are 18 bits wide. Consequently, 2 buffer locations are used to store these values. 12 bits are stored in the first buffer, and the remaining 6 bits are stored in the 8 lower bits of the second. The upper 4 bits of the second buffer are unused.

Bits 4 and 8 of all I/O addresses are swapped by the processor board. Logic analyzers which monitor the terminal's bus will not see the same addresses as used by the processor.

When transferring data to the 8-bit wide registers, ALL flags in the register are set to the values transferred. It is not possible to change just one of the flags in a register. Consequently, the flags used by the 'mode' and 'flags' registers (described later in this section) are stored in RAM by the 8080. When one bit needs to be changed, the current state of all bits is loaded, the appropriate bit is altered, then all are sent to the hardware.

7.1.5 ADDRESSES

The following table summarizes the registers, buffer locations and strobes available to the 8080. The buffer value, such as B(14) is the notation used by the graphics controller documentation. The symbolic name is that used in the firmware listing to reference a specific address. If more than one symbol is given for an address, the buffer location is used for different purposes at different times.

HARDWARE USE	ADDRESS	SYMBOL	USE
STATUS	104440B	HWSTAT	READ CONTROLLER STATUS
STROBES			
HARD RESET STROBE	104540B	GRESET	RESET CONTROLLER
RETRACE STROBE	104541B	VRESET	RESET VERTICAL RETRACE FLAG
REGISTERS			
FLAGS	104440B	HWFLGS	LOAD FLAGS F1,F3,F4,F5
MODE	104501B	HCEJK	LOAD DRAWING MODE
PATTERN	104500B	PATERN	LOAD DOT-DASH PATTERN
PATTERN SCALE	104441B	SCALER	LOAD PATTERN SCALE FACTOR

BUFFER LOCATIONS				
B(0)	104436B	D1	DISCRIMINANT INCREMENT #1	
		GC1DC	HORIZONTAL CURSOR DOT CNT	
B(1)	104434B	D2	DISCRIMINANT INCREMENT #2	
		GC2DC	VERTICAL CURSOR DOT CNT	
B(2)	104432B	M1	12 LSB MEMORY INCREMENT #1	
		GC1LO	12 LSB HORIZ CURSOR ADDRESS	
B(3)	104430B	SIGNM1	6 MSB MEMORY INCREMENT #1	
		GC1HI	6 MSB HORIZ CURSOR ADDRESS	
B(4)	104426B	M2	12 LSB MEMORY INCREMENT #2	
		GC2LO	12 LSB VERT CURSOR ADDRESS	
B(5)	104424B	SIGNM2	6 MSB MEMORY INCREMENT #2	
		GC2HI	6 MSB VERT CURSOR ADDRESS	
B(6)	104422B	DC	DOT COUNT	
		ZALO	12 LSB ZOOM ADDRESS	
B(7)	104420B	INITD	INITIAL DISCRIMINANT	
		ZAHI	6 MSB ZOOM ADDRESS	
	104421B	MSBD	4 MSB OF DISCRIMINANT D	
B(8)	104416B	LSBWA	12 LSB OF VECTOR ADDRESS	
B(9)	104414B	MSBWA	6 MSB OF VECTOR ADDRESS	
B(10)	104413B	SELWA	FLAG--SELECT OLD/NEW VECTOR ADDRESS (MSB OF B(10))	
B(11)	104410B	ZOOMRC	ZOOM REPEAT COUNT	
	104411B	SLFTST	FLAG--START SELF TEST (MSB OF B(11))	
B(12)	104406B	ZOOMWC	ZOOM WORD COUNT	
	104407B	CONTST	FLAG--CONTINUE SELF TEST (MSB OF B(12))	
B(13)	104404B	DCNTRL	ZOOM DISPLAY CONTROL BYTE (8 BITS ONLY)	
	104405B	PRESHF	ZOOM PRESHIFT (4 BITS ONLY)	
B(14)	104402B	VDC	VECTOR DRAWING DOT COUNT	
B(15)	104401B	DRWDOT	FLAG--DRAW FIRST DOT OF VECTOR (MSB OF B(15))	

7.2 HARDWARE STATUS

A read from the location 104440B returns 8 bits of status from the graphics hardware.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	SELF	RE-	BAD	BAD	BAD	BAD	
DATA	TEST	TRACE	PACK	PACK	PACK	PACK	BUSY
	FAIL	FLAG					

BUSY--when the controller has been loaded with vector, cursor, or zoom parameters, and the appropriate flags have been set (F1 or F4, described later in this section) the hardware is busy until the operation is completed. The 8080 cannot access any buffer locations while this bit is set.

BAD PACK--These 4 bits indicate which of the 16 RAM chips failed if the self test flag is set during self test (see Section 7.9).

RETRACE--This bit is set by the hardware at the beginning of vertical retrace, which occurs once each frame. It must be cleared by a strobe from the 8080 (Section 7.3).

SELF TEST FAIL--When set, this bit indicates that a failure occurred during self test (Section 7.9).

DATA--When drawing a vector, the state of each memory bit in the vector (before being altered) is stored here. The last bit of the vector is therefore accessible. By drawing vectors one dot long, it is possible to read the state of every bit in the image memory (Section 7.10).

7.3 STROBES

The two strobes used are generated by any write operation to the indicated address. The data is ignored.

GRESET--The graphics controller is not reset by the power on signal generated by power on or a hard reset. Instead, this strobe must be explicitly sent by the firmware to initialize the graphics hardware.

VRESET--This strobe clears the vertical retrace flag, which is set by the hardware.

7.4 REGISTERS

7.4.1 FLAGS REGISTER

Four flags are used to tell the controller what the parameters in the buffer locations are to be used for.

BIT 4		BIT 3		BIT 1		BIT 0				
			F5		F4		F3		F1	

F1--This flag is set to indicate that the buffer contains either vector or cursor data. It initiates vector generation. It is referenced by the equate BUSY.

F3--This flag is set to turn zoom mode on. (ZOOM)

F4--This flag is set to indicate that the buffer contains new zoom parameters. New parameters must be loaded when zoom is initially turned on, or when the zoom size or position changes. (NEWZM)

F5--This flag is set to indicate that the vector data loaded is to be used to draw a cursor. (DRWGC)

To draw a vector, the proper parameters are loaded (Section 7.6) and flag F1 is set. Flags F4 and F5 must be cleared. Flag F3 will be set if zoom mode is on.

To draw a cursor, the cursor parameters are loaded (Section 7.7) and both flags F1 and F5 are set. Flag F4 must be cleared, and flag F3 will be set if zoom mode is on.

To turn zoom mode on, or to change zoom size or position, the zoom parameters are loaded (Section 7.8) and flags F3 and F4 are set. Flags F1 and F5 must be cleared.

Flags F1 and F4 are cleared when the vector or zoom parameters have been processed by the hardware. If either is set, the 'BUSY' bit will be returned by the hardware status. This BUSY flag must be tested before sending any parameters to the hardware to insure that the previous operation has been completed.

7.4.2 MODE REGISTER

The mode register controls the way in which the image memory is modified when vectors are drawn. Bits can be set, cleared, or complemented with or without a pattern.

	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0

		S	H	C	E	J
						K

EJK--The EJK bits control the drawing mode. Whenever an image memory bit is to be modified, the operation that occurs is determined by the EJK (and possibly pattern) bits.

E	J	K	OPERATION
0	0	0	Do nothing (leave memory unchanged).
0	0	1	Clear the bit always.
0	1	0	Set the bit always.
0	1	1	Complement the bit always.
1	0	0	Copy the pattern bit (jam pattern).
1	0	1	Clear the bit if the pattern bit is set, otherwise do nothing.
1	1	0	Set the bit if the pattern bit is set, otherwise do nothing.
1	1	1	Complement the bit if the pattern bit is set, otherwise do nothing.

C--This bit, when set, initiates a clear or set memory operation (Section 7.5).

H--This bit, when cleared, inhibits the graphics video output. when set, the graphics video is turned on.

S--This bit has two functions. During self test (Section 7.9) it is used as a sample against which every bit in the memory is compared.

When this bit is cleared, the pattern and pattern scale registers are prevented from changing, as they normally do after each write operation. This is used to prevent the cursor or rubber band line from changing the state of the pattern.

7.4.3 PATTERN AND PATTERN SCALE REGISTERS

These registers are used to generate dotted and dashed lines and graphics text characters. An 8 bit prototype pattern is loaded into the pattern register. Assuming a scale factor of 0, each time a dot is drawn the pattern is shifted. If the drawing mode has enabled the pattern, the pattern bit is used to determine what is actually written into the memory. The first pattern dot used is the MSB (bit 7). When the pattern is shifted it is rotated to the left, so that bit 6 is used next, then bit 5, and so on. If the drawing mode were set to 'jam pattern', the vector drawn would be made up of copies of the pattern, repeating every 8 bits. If the scale factor is set to a non zero value, then each pattern bit is used that many times before shifting. Each bit in the pattern would then correspond to more than one bit on the screen, effectively stretching the pattern.

The pattern register (PATTERN) is loaded with the 8 bit pattern. The scale register (SCALER) is loaded with a 4 bit scale factor (0-15) which corresponds to the factor 1X to 16X. Remember that the S bit in the mode register must be set to enable the pattern shift.

7.5 CLEARING AND SETTING THE SCREEN

The graphics image memory is cleared or set by loading the proper mode bits and asserting the C bit. As each memory word is read to display the graphics image, every bit in it is set to 0 or 1, respectively. Consequently, it takes one frame time to clear or set the entire screen. As noted in Section 5.2.1, when zoomed more of the memory than is being displayed may be cleared. It is not possible to complement the image memory this way, since the same data is written into all bits of each 16 bit word.

When the C bit is turned off after the clear is finished, only the C bit should be changed, or spurious data may be written into the memory. The graphics video should be turned off during the clear, since the outputs from the memory chips is undefined while they are being written into. The procedure for setting or clearing the memory is as follows.

1. Set the HCEJK bits as follows:
To clear, C=1, K=1, all others 0.
To set, C=1, J=1, all others 0.
2. wait at least 1 complete frame. This is most easily done by using the real-time timer to wait 20 milliseconds.
3. Clear the C bit to 0, but maintain the others as they were in step 1.
4. Restore the SHCEJK bits to what they were before the clear operation.

7.6 VECTOR GENERATION

To generate a vector, the parameters described in Section 7.1.3 must be computed and set to the appropriate buffer locations. In addition, there are several other locations not described in the algorithm which must be properly loaded.

7.6.1 ALGORITHM PARAMETERS

After insuring that the controller is idle, the following buffer locations are loaded with the parameters described in Section 7.1.3. Note that certain values are stored as the negative of the actual value. Negative numbers are represented in two's complement form.

PARAMETER	SYMBOLIC ADDRESS
INITIAL DISCRIMINANT	D
DISCRIMINANT INCREMENT #1	D1
DISCRIMINANT INCREMENT #2	D2
VECTOR LENGTH (STORED AS -LENGTH)	DC
INITIAL START ADDRESS (18 BITS) (NEED NOT BE SENT EVERY TIME, SEE SECTION 7.6.2)	LSBWA (0-11) MSBWA (12-17)
MEMORY INCREMENT #1 (18 BITS)	M1 (0-11) SIGNM1 (12-17)
MEMORY INCREMENT #2 (18 BITS)	M2 (0-11) SIGNM2 (12-17)

The memory increments M1 and M2 are determined by the signs of Delta X, Delta Y, and $|\Delta X| - |\Delta Y|$

SIGN OF DELTA X	SIGN OF DELTA Y	SIGN OF $ \Delta X $ - $ \Delta Y $	OCTANT	M1	M2
+	+	+	1	+1	-719
+	+	-	2	-720	-719
+	-	+	8	+1	+721
+	-	-	7	+720	+721
-	+	+	4	-1	-721
-	+	-	3	-720	-721
-	-	+	5	-1	+719
-	-	-	6	+720	+719

7.6.2 OTHER PARAMETERS

The following buffer location must be loaded with additional parameters required by the hardware.

1. SELECT OLD/NEW ADDRESS FLAG, BUFFER B(10)--If a series of connected vectors is being drawn, the endpoint of the previous vector is the same as the starting point of the next vector. Consequently, the hardware will already have the vector starting address. Since it takes a significant amount of time to do the conversion from X,Y to the 18 bit representation, it is advantageous to tell the hardware to use the address it has when it is possible, rather than always explicitly sending the address. To tell the hardware to use the old address it already has, a value of 0 is set to location SELWA. To use a new address, it is sent to the location LSBWA and MSBWA, and a value of 10B is set to location SELWA.
2. SELF TEST FLAGS, BUFFERS B(11) AND B(12)--Two flags control the operation of self test. They must be set to 0 before drawing a vector. Since these same locations are also used for zoom data, these locations should be cleared before every vector. This is done by sending 0 to locations SLFTST and CONTST.
3. VECTOR DRAWING DOT COUNT, BUFFER B(14)--The controller can generate vector dots only after it has read the data required to generate a scan line. Only a portion of the vector can be drawn before the controller must stop drawing and read the data for the next scan line. This buffer location contains the number of dots to be drawn between read operations. For normal operation, at most 4 dots can be drawn. In zoom mode, at most 3 dots can be drawn. If it is not necessary to generate the display, as when the graphics video is off, a maximum of 250 dots can be specified. As noted in Section 5.2.2, there is a substantial speed improvement if this is done. If more than 250 dots are specified, or a value greater than 3 while in zoom mode, the dynamic memories may not be refreshed, which will destroy the image.

The values are sent as the negative of the true value to the location VDC. To summarize,

NORMAL MODE	VDC = -4
ZOOM MODE	VDC = -3
HIGH SPEED	VDC = -250

4. DRAW FIRST DOT FLAG, BUFFER B(15)--When connected vectors are drawn, the first dot of the new vector is the same as the last dot of the previous vector. Ordinarily, drawing the dot twice does not matter, but if complement mode or a line pattern is used, drawing the dot twice will cause problems. Setting this flag, by storing a value of 10B in location DRWDOT will cause the first dot of the vector to be drawn. If 0 is stored, the first dot of the vector will not be drawn.

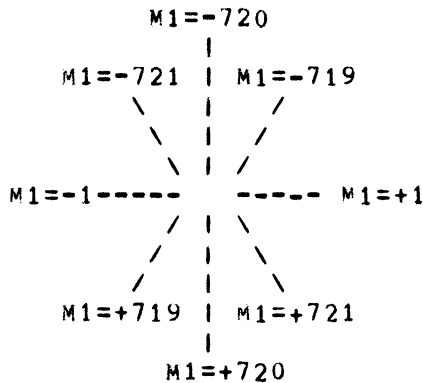
In addition to the buffer locations, the mode and pattern registers should be set to the desired state. Note that the pattern and scale values should not be changed for every vector when connected lines are drawn, or the pattern will be reset for each vector.

The cursor and rubber band line must be turned off before anything is drawn in the graphics memory (Section 7.7).

Once the parameters have been loaded, flag F1 is set as indicated in Section 7.4.1 to start the vector generator.

7.6.3 HORIZONTAL AND VERTICAL VECTORS

It is possible to draw horizontal, vertical, or diagonal vectors by specifying fewer parameters. As can be seen from the vector algorithm, if the discriminant value never changes, the same memory increment will always be added. If the discriminant D is initially set to any negative value, and the discriminant increment D1 is set to 0, then memory increment M1 will always be added to obtain the vector points. By selecting M1 as follows, a vector can be drawn in any of the 8 indicated directions.



To draw a single dot, even fewer parameters are needed. The only algorithm parameters needed are the address and a dot count of -1. The other flags such as draw first dot and the self test flags must also be set to the proper values.

7.6.4 FIRMWARE ROUTINES

The following routines in the microcode pertain to vector generation.

WAIT--WAIT FOR IDLE CONTROLLER

ENTRY DON'T CARE

EXIT ALL REGISTERS SAVED

This routine loops until the busy bit is cleared, indicating the controller is ready for more parameters.

MPY45--COMPUTE (359-HL)*45

ENTRY HL = Y COORDINATE

EXIT HL = (359-Y)*45

This routine performs part of the conversion from X, Y coordinates to an 18 bit address.

GETWA--COMPUTE 18 BIT ADDRESS

ENTRY HL = (359-Y)*45 (FROM MPY45)
DE = X COORDINATE

EXIT HL = 12 LSB OF ADDRESS
A = 6 MSB

This routine is used to compute the actual 18 bit address from the input values. An address conversion typically appears as follows:

LHLD	YCOORD	GET THE Y COORDINATE
CALL	MPY45	COMPUTE (359-Y)*45
XCHG		DE = Y VALUE
LHLD	XCOORD	FETCH THE X COORDINATE
XCHG		HL=Y, DE=X
CALL	GETWA	CONVERT TO 18 BITS
CALL	WAIT	INSURE THE CONTROLLER IS IDLE
SHLD	LSBWA	SEND BITS 0-11
STA	MSBWA	SEND BITS 12-17

SETUP--COMPUTE DELTA X, DELTA Y, BOUNDS CODES, OCTANT

ENTRY XCURR, YCURR = VECTOR STARTING POINT
XNEW, YNEW = ENDING POINT

EXIT DELTAX, DELTAY, OCTANT UPDATED

This computes the parameters needed to determine the algorithm parameters. The location OCTANT upon exit contains the signs of Delta X and Delta Y. DELTAX and DELTAY contain the absolute values of Delta X and Delta Y. The rest of the computation is done in the routine DRWVEC.

DRWVEC--COMPUTE ALGORITHM PARAMETERS AND SEND TO HARDWARE

ENTRY--OCTANT, DELTAX, DELTAY AS ABOVE
XSTART, YSTART CONTAINS STARTING POINT

This routine uses DELTAX and DELTAY to compute D, D1, D2, etc. It uses the sign bits in OCTANT as well as the sign of DELTAX-DELTAY to determine the M1 and M2 values. If necessary the starting address is sent. The value in XSTART, YSTART is used since the clipping routine may have changed the starting point of the vector.

HLINE--DRAW HORIZONTAL VECTOR (LEFT TO RIGHT)

VLINE--DRAW VERTICAL VECTOR (BOTTOM TO TOP)

ENTRY DE = X COORDINATE
HL = Y COORDINATE
BC = -(LENGTH)

This routine draws constant direction vectors.

Other routines of possible interest are ABFILL, which fills a rectangular area with horizontal vectors, and CHFILL, which draws graphics character images.

7.7 GRAPHICS CURSOR

The graphics cursor is drawn as intersecting horizontal and vertical vectors. The firmware determines the starting point for each vector so that the point of intersection is in the proper place, and the vector length for each vector, so that the cursor does not extend off screen. The two addresses and lengths are send to the controller, which draws the two vectors during vertical retrace, while the screen is blanked. This insures that partially drawn cursors will not be displayed. To move the cursor, the lines at the old position must be erased, then new lines drawn. If the cursor were erased by clearing the image memory bits, gaps would be left in any line the cursor intersected. Large parts of the display would be erased as the cursor moved across the screen. Consequently, the cursor must be initially drawn in complement mode. It is erased by recomplementing the same bits. Complementing a bit twice restores it to its original state. Complementing also insures that the cursor will always be visible, regardless of the background. However, gaps will now appear in intersecting vectors when the cursor is drawn. To remedy this, the cursor is complemented every frame. The resulting cursor appears half bright, since it is only visible every other frame, but it does not cause gaps when placed on other vectors.

For this complement/recomplement scheme to restore the display when the cursor is removed, no vectors that intersect the cursor can be drawn while the cursor is on. Otherwise, when the cursor was turned off the intersecting bits in the vector would also be complemented, leaving gaps. Consequently, the cursor is turned off before anything is written into the image memory.

To draw the cursor, the following parameters are loaded. Note that the horizontal vector is draw from left to right, and the vertical vector from bottom to top. The starting address must be compensated by one point. The horizontal address must have 1 subtracted from the 18 bit value, and the vertical address must have 720 added to the 18 bit value.

BUFFER	VALUE
GC1DC	NEGATIVE OF HORIZONTAL VECTOR LENGTH
GC2DC	NEGATIVE OF VERTICAL VECTOR LENGTH
GC1LO	12 LSB OF HORIZONTAL VECTOR COMPENSATED ADDRESS
GC1HI	6 MSB OF HORIZONTAL VECTOR COMPENSATED ADDRESS
GC2LO	12 LSB OF VERTICAL VECTOR COMPENSATED ADDRESS
GC2HI	6 MSB OF VERTICAL VECTOR COMPENSATED ADDRESS

The drawing mode should be set to complement, and the S bit set to 0 to prevent the pattern from changing. Flags F1 and F5 are set to indicate that a cursor is to be drawn.

7.7.1 FIRMWARE ROUTINES

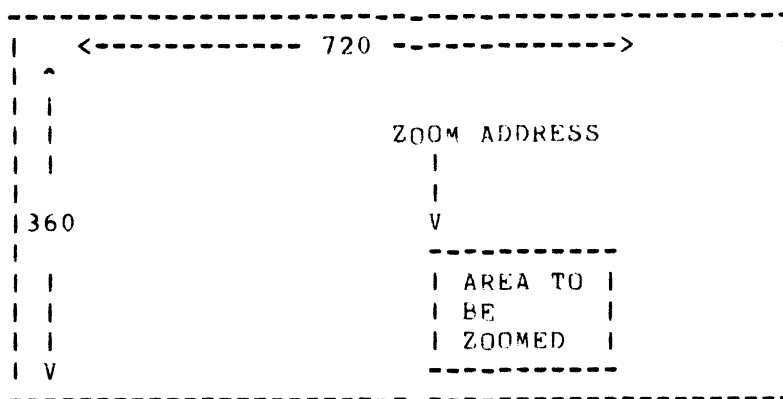
The following microcode routines pertain to cursor generation.

1. SUPRGC--Suppresses the cursor before drawing a vector. The state of the cursor is saved, so that it can be turned back on if necessary when the vector is finished. The cursor can be suppressed by one of several flags, all of which must be cleared before the cursor will be drawn. For example, a suppress flag is set when the graphics video is turned off. Another flag is set when a vector is drawn. When the vector is finished, the second flag will be cleared but the cursor will remain off until the first is also when the video is turned back on. The flag TMSUP is automatically cleared after a preset time interval. This is so that the cursor does not have to be turned off, then on after every vector.
2. ENABGC--Re-enables the cursor after being suppressed.
3. DRAWGC--Computes the parameters for the cursor and sends them to the controller.
4. GCMON--Monitors the cursor keys and updates the cursor position.

7.8 ZOOM

To cause the display to zoom, the microprocessor must send the following parameters to the hardware.

1. ZOOM ADDRESS, BUFFERS ZALO AND ZAH1--This is the 18 bit address of the upper left hand corner of the area to be zoomed. Zoom can start at any bit in the memory.



2. REPEAT COUNT, BUFFER ZOOMRC--This is the negative of the zoom magnification minus 1. Zoom sizes 2 through 16 are given by values of -1 through -15. It tells the hardware how many times each scan line is to be repeated.
3. WORD COUNT, BUFFER ZOOMWC--This parameter tells the controller how many 16 bit image memory words must be read to generate one scan line. At 1X, 45 words are read (45 words X 16 bits/word = 720 bits). For zoom size M, the number of words is approximately 45/M. This is not quite right, because if the zoom address is in the middle of a word, an extra word may have to be read. Given the X coordinate of the center of the zoom area (not the zoom address), the word count is obtained by:

LEFTMOST WORD = $(X - (360/M)) / 16$
RIGHTMOST WORD = $(X + (360/M)) / 16$
WORD COUNT = XRIGHT - XLEFT + 1

ZOOMRC = -WORD COUNT (SENT TO HARDWARE AS NEGATIVE OF VALUE)

4. PRESIFT, BUFFER PRESHF--The hardware can only read the memory 16 bits at a time. If the zoom address starts in the middle of a 16 bit word, the leading bits, from 0 to 15, must be discarded. This 4 bit value tells how many leading bits are unused. It is sent as a 1's complement value. To compute this, take the 4 LSB of the zoom address and complement them.
5. DISPLAY CONTROL BYTE, BUFFER DCNTRL--This 8 bit value controls how wide a dot is when in zoom mode. For zoom size M, the hardware generates M dots for each dot in the image memory. Then, the last dot is blanked. This can be changed to display all dots, deleting the blank at the end, or to blank more than one dot. The values used are as follows.

ZOOM SIZE	CONTROL BYTE	ZOOM SIZE	CONTROL BYTE
2 X	357B	10 X	147B
3 X	336B	11 X	126B
4 X	315B	12 X	105B
5 X	274B	13 X	064B
6 X	253B	14 X	043B
7 X	232B	15 X	022B
8 X	211B	16 X	001B
9 X	170B		

To change the number of dots displayed, the control byte is built up from 2 values.

BIT 7	BIT 4	BIT 3	BIT 0

TWO'S COMPLEMENT OF	TWO'S COMPLEMENT OF THE		
MAGNIFICATION	NUMBER OF DOTS TO BE		
(-2 THRU -16)	DISPLAYED (-1 THRU -16)		

For example to zoom 16X and display 15 dots, the values are:

-MAGNIFICATION = -(10000) = 10000 => 0000
-(DISPLAY 15 DOTS) = -(1111) = 0001
DISPLAY CONTROL BYTE = 0018

6. VECTOR DRAWING DOT COUNT, BUFFER VDC--This must be set to -3 when zoomed.

When the buffer is loaded, zoom is turned on by setting flags F3 and F4. Flag F4 is cleared by the controller when the zoom parameters have been processed. Zoom is turned off when the microprocessor clears flag F3. Zoom is actually turned on or off only between frames during vertical retrace.

7.8.1 FIRMWARE ROUTINES

Relevant routines in the microcode are ZMUPDA, which computes the zoom parameters from the cursor location, NWSIZE and UNZOOM, which control zoom size, and VR, which does zoom, cursor and rubber band line updates once per frame.

7.9 SELF TEST

Self test compares the state of the image memory against a sample bit as vectors are drawn. If a data bit differs from the sample, the vector being drawn is stopped, an error flag is set, and the pack number of the 16 K memory chip which contains the bad bit is noted. This allows the 8080 to put the image memory in a known state, then test every bit to see if it responded correctly.

To initiate self test, the parameters for a vector are loaded, as for a normal vector. The self test flag is set. The memory is set or cleared, and the S bit in the mode register is set accordingly. If the controller finds a mismatch as it draws the vector, drawing is stopped and the bad pack noted. The continue self test flag allows vector generation to be re-started from the point of failure.

The sequence of events in the 2648A test are:

1. Clear the screen, set S bit to 0.
2. Turn on complement mode.
3. Draw vertical vectors bottom to top, starting at X=0 and moving to X=719.
4. The screen should now be completely complemented (all bits set), set S bit to 1.
5. Draw vertical vectors top to bottom, starting at X=719 and moving to X=0.
6. The screen should now be clear again, set S bit to 0.
7. Draw horizontal vectors left to right, starting at Y=0 and moving to Y=359.
8. The screen should now be set, set S bit to 1.
9. Draw horizontal vectors right to left, starting at Y=359 and moving to Y=0.
10. The screen is then set, and steps 2-9 repeated with the opposite sense of data (set now where clear before and vice-versa).

To start self test:

1. The vector parameters are loaded as before.
2. The self test flag is set by storing 10B in buffer SLFIST.
3. The continue self test flag is cleared by storing 0 in buffer CONTST.
4. The S bit in the mode flags is put into the proper state.
5. Flag F5 in the flags register is cleared, and the start vector flag F1 is set.

When the busy bit in the status byte indicates the vector is finished, flag F5 is tested. If 0, the memory compared successfully. If set, then an error occurred. The bad pack bits are interpreted as follows:

BIT	4	3	2	1	PACK	BIT	4	3	2	1	PACK
	0	0	0	0	U 11		1	0	0	0	U 12
	0	0	0	1	U 21		1	0	0	1	U 22
	0	0	1	0	U 31		1	0	1	0	U 32
	0	0	1	1	U 41		1	0	1	1	U 42
	0	1	0	0	U 51		1	1	0	0	U 52
	0	1	0	1	U 61		1	1	0	1	U 62
	0	1	1	0	U 71		1	1	1	0	U 72
	0	1	1	1	U 81		1	1	1	1	U 82

6. To continue the test from the point of failure, the continue self test flag is set by storing 10B in buffer location CONTST. The error flag F5 is cleared, and the start vector flag F1 is set.

Note that flag F5 is used for two purposes, to signal a self test error (when set by the controller) and to indicate that a cursor is to be drawn (when set by the 8080). Flag F5 must be cleared before drawing a vector or it will always be interpreted as draw cursor.

7.9.1 FIRMWARE ROUTINES

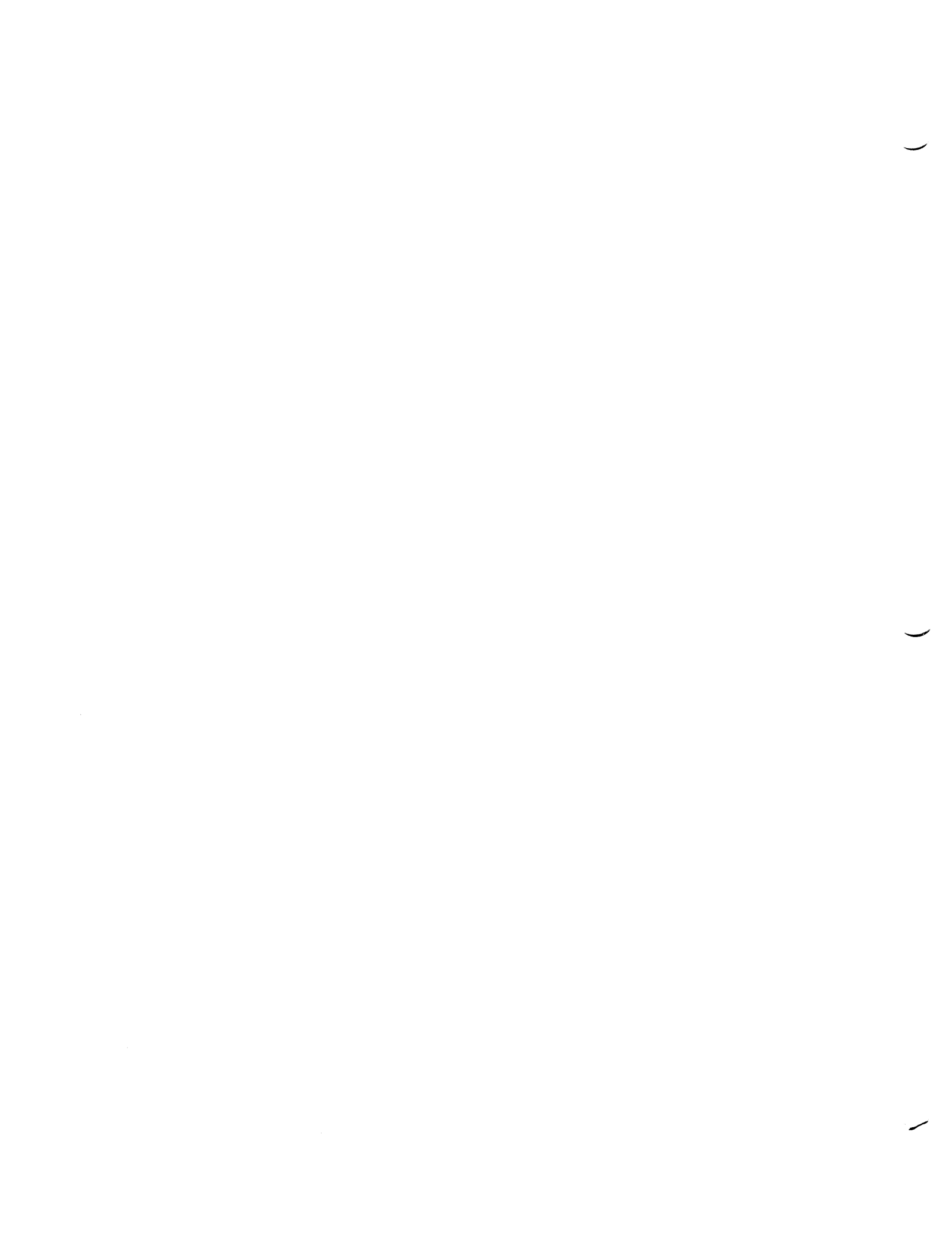
Relevant routines pertaining to self test are HTEST and VTEST, which draw the vertical and horizontal lines, STDRAW, which initiates vector drawing when in self test, and STFAIL, which is called in the event of an error when testing.

7.10 READING THE IMAGE MEMORY

The 8080 can read the data stored in the image memory one bit at a time. The status byte contains the bit read before the last vector endpoint was drawn. Drawing a one dot vector will return the value of that bit. The drawing mode should be set to 'Do Nothing' to prevent the image memory from being changed as it is read.

To read an image memory bit, the following values must be sent:

1. The 18 bit address, buffers WALD and WAHL.
2. A dot count of -1, buffer DC.
3. The draw first dot flag, 10B in buffer DRWDOT.
4. The use new address flag, 10B in buffer SELWA.
5. Self test off, 0 in buffers SLFTST and CONTST.
6. Set the drawing mode to do nothing.
7. Set flag F1 to initiate the read.
8. When the busy bit in the status byte is 0, the data bit contains the value at that memory location



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE  1
=====
  1      0000      . . .          ASB,HEX                ;PT91 17AUG77
  2      0000      . . .          ;*****
  3      0000      . . .          ; THIS IS THE ROM CODE PT90 OF 21JUN77 WITH ONE
  4      0000      . . .          ; ROM MODIFIED.  THE ROUTINE 'STTERM' IN LOCATIONS
  5      0000      . . .          ; 14322 TO 14431 WAS MODIFIED FOR MULTIPOINT
  6      0000      . . .          ; COMPATIBILITY.
  7      0055      . . .          VERSN2 EQU 1250          ;NEW ROM = VERSION 'U'
  8      0000      . . .          ;*****
  9      0000      . . .          ;*****
 10     0000      . . .          ; 2645 MAIN CODE MODIFIED FOR GRAPHICS
 11     0000      . . .          ;*****
 12     0000      . . .          ;*****
 13     0000      . . .          ; VERSION LEVEL CODE *
 14     0000      . . .          ;*****
 15     0000      . . .          ;*****
 16     0054      . . .          VERSN EQU 1240          ;GRAPHICS = VERSION 'T'
 17     0000      . . .          ;*****
 18     0000      . . .          ;
 19     0000      . . .          ; COMMON EQUATES - CM34 - 6/10/76 - 1315 HRS.
 20     0000      . . .          ;
 21     9100      . . .          FSTRAM EQU 1104000      ;FAST RAM LOWER LIMIT
 22     0000      . . .          ;*****
 23     0000      . . .          ; KBDCSW - KEYBOARD DATA COMM SWITCHES *
 24     0000      . . .          ;*****
 25     0080      . . .          FULDUP EQU 2000        ;HALF/FULL DUPLEX
 26     0000      . . .          ;*****
 27     0000      . . .          ; KBJMPR - KEYBOARD INTERFACE JUMPERS *
 28     0000      . . .          ;*****
 29     0000      . . .          ;
 30     0000      . . .          ; JUMPERS SENSED AS 0' WHEN INSERTED
 31     0000      . . .          ;
 32     0000      . . .          ; ALL JUMPERS ARE NORMALLY INSERTED
 33     0000      . . .          ;
 34     0001      . . .          CONDIS EQU 0010        ;CONTROL CODE DISABLE
 35     0000      . . .          ; (0=DISABLED)
 36     0002      . . .          SPLDIS EQU 0020        ;SPOW LATCH DISABLE
 37     0000      . . .          ; (0=DISABLED)
 38     0004      . . .          LINWRP EQU 0040        ;COLUMN 80 AUTO CR,LF
 39     0000      . . .          ; (0=ENABLED)
 40     0008      . . .          PAGSTR EQU 0100        ;PAGE MODE STRAP
 41     0000      . . .          ; (0=LINE-FIELD MODE)
 42     0010      . . .          LFPOS EQU 200          ;LINE FEED POSITION
 43     0000      . . .          ; (0 = POSITION LINE FEED
 44     0000      . . .          ; AT START OF NEXT I/O
 45     0000      . . .          ; READ
 46     0000      . . .          ; 1 = PUT LINE FEED AT END
 47     0000      . . .          ; OF RECORD)
 48     0020      . . .          FSTSND EQU 400         ;9600 BAUD DATACOM SHIFT
 49     0000      . . .          ; (0=9600 BAUD FOR ESC,E)
 50     0040      . . .          HNSHKE EQU 1000        ;BLUCK TRANSFER HANDSHAKE
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                PAGE  2
=====
 51      0000      . . .      ;                               (0 = FOLLOW DC2SND SETTING
 52      0000      . . .      ;                               1 = SEND DC2 BEFORE DATA)
 53      0080      . . .      DC2SND EQU 2000
 54      0000      . . .      ;                               (0 = SEND DC2 ON ENTER
 55      0000      . . .      ;                               AND FUNCTION KEY IN
 56      0000      . . .      ;                               BLOCK MODE
 57      0000      . . .      ;                               1 = INHIBIT ALL DC2
 58      0000      . . .      ;                               HANDSHAKE)
=====
```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE   3
=====
 60      0000      . . .      ;*****
 61      0000      . . .      ; KBJMP2 - SECOND SET OF KEYBOARD JUMPERS *
 62      0000      . . .      ;*****
 63      0001      . . .      AUTTRM EQU 1Q          ;AUTO TERMINATE ON "ENTER"
 64      0002      . . .      CLRTRM EQU 2Q          ;CLEAR TERMINATOR ON TRANSMI
 65      0004      . . .      NOTEST EQU 4Q          ;INHIBIT TERMINAL SELF-TEST
 66      0008      . . .      EDTWRP EQU 10Q         ;INVERT SENSE OF EDIT WRAP
 67      0010      . . .      PRNLTA EQU 20Q         ;SEND ALL CODES TO PRINTER
 68      0080      . . .      DCJMP0 EQU 200Q        ;DATA COMM JUMPER
 69      0000      . . .      ;*****
 70      0000      . . .      ; KBJMP3 - THIRD SET OF KEYBOARD JUMPERS *
 71      0000      . . .      ;*****
 72      0001      . . .      DCJMP1 EQU 1Q          ;DATA COMM JUMPERS
 73      0002      . . .      DCJMP2 EQU 2Q          ;.
 74      0004      . . .      DCJMP3 EQU 4Q          ;.
 75      0008      . . .      DCJMP4 EQU 10Q         ;.
 76      0010      . . .      NUDCST EQU 20Q         ;INHIBIT DATA COMM SELF-TEST
 77      0000      . . .      ;              (0 = DISABLED)
 78      0020      . . .      SETCH EQU 40Q          ;TURN ON "CH" CONTROL LINE
 79      0000      . . .      ;              (0 = OFF, 1 = ON)
 80      0040      . . .      CHEKCC EQU 100Q        ;MONITOR CC CONTROL LINE
 81      0000      . . .      ;              (1 = ENABLED)
 82      0080      . . .      FRCPTY EQU 200Q        ;FORCE PARITY ON/NO IN CHECK
 83      0000      . . .      ;              (1 = ENABLED)
 84      0000      . . .      ;*****
 85      0000      . . .      ; CMFLGS - COMMON FLAGS *
 86      0000      . . .      ;*****
 87      0001      . . .      BLKTRG EQU 1Q          ;BLOCK TRANSFER TRIGGER
 88      0002      . . .      INSWRP EQU 2Q          ;INSERT WITH WRAP AROUND
 89      0004      . . .      FRCRST EQU 4Q          ;FORCE FULL TERMINAL RESET
 90      0008      . . .      DEFSKY EQU 10Q         ;DEFINE SOFT KEY MODE ENABLE
 91      0010      . . .      REMSET EQU 20Q         ;REMOTE MODE ENABLED
 92      0020      . . .      RCVMD E EQU 40Q        ;TERMINAL IN RECEIVE MODE
 93      0000      . . .      ;*****
 94      0000      . . .      ; ERRFLG - ERROR FLAGS *
 95      0000      . . .      ;*****
 96      0001      . . .      DCMERK EQU 1Q          ;DATACOM (1 = ERROR)
 97      0002      . . .      TESTOK EQU 2Q          ;SELF-TEST (0 = ERROR)
 98      0004      . . .      LDRCHK EQU 4Q          ;LOADER CHECKSUM (0 = ERROR)
 99      0000      . . .      ;*****
100      0000      . . .      ; INTFLG - INTERRUPT FLAG *
101      0000      . . .      ;*****
102      0003      . . .      TMRINT EQU 3           ;TIMER INTERRUPT
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
104	0000	.	.	;*****	4
105	0000	.	.	; PRCCTL - PROCESSOR CONTROL FLAGS *	
106	0000	.	.	;*****	
107	0000	.	.	TMIACK EQU 00 ;ACKNOWLEDGE TIMER INTERRUPT	
108	0000	.	.	; (BIT 1 OFF)	
109	0001	.	.	TMRON EQU 10 ;SET TIMER ON	
110	0002	.	.	TMIEN EQU 20 ;RE-ENABLE TIMER INTERRUPT	
111	0010	.	.	DCIOFF EQU 200 ;DISABLE DATA COMM INTERRUPT	
112	0020	.	.	TMIOFF EQU 400 ;DISABLE TIMER INTERRUPTS	
113	0040	.	.	POLL EQU 1000 ;POLL CTU INTERRUPTS	
114	0000	.	.	;V*V*V*V* SET TO ZERO FOR ROM VERSION *V*V*V*V*	
115	0000	.	.	SETROM EQU 00 ;0 => ENABLE ROM	
116	0000	.	.	;*****	
117	0000	.	.	; MDFLG1 - TERMINAL MODE FLAGS 1 *	
118	0000	.	.	;*****	
119	0001	.	.	DSPFNC EQU 10 ;DISPLAY FUNCTIONS ENABLED	
120	0002	.	.	INSCHR EQU 20 ;INSERT CHARACTER ENABLED	
121	0004	.	.	MEMLOK EQU 40 ;MEMORY LOCK ENABLED	
122	0008	.	.	FORMAT EQU 100 ;FORMAT MODE ENABLED	
123	0010	.	.	EDIT EQU 200 ;EDIT MODE ENABLED	
124	0020	.	.	SELECT EQU 400 ;SELECT MODE ENABLED	
125	0040	.	.	RECORD EQU 1000 ;RECORD MODE ENABLED	
126	0080	.	.	FORGN EQU 2000 ;FOREIGN MODE ENABLED	
127	0000	.	.	;*****	
128	0000	.	.	; MDFLG2 - TERMINAL MODE FLAGS 2 *	
129	0000	.	.	;*****	
130	0001	.	.	CAPSLK EQU 10 ;CAPS LOCK ENABLED	
131	0002	.	.	BLKMDE EQU 20 ;BLOCK MODE ENABLED	
132	0004	.	.	AUTOLF EQU 40 ;AUTO LF ENABLED	
133	0008	.	.	REMOTE EQU 100 ;REMOTE ENABLED	
134	0020	.	.	WBSK EQU 400 ;WRITE-BACKSPACE-READ MODE	
135	0000	.	.	;*****	
136	0000	.	.	; RADIX - BASE OF INPUT PARAMETER FOR ESC SEQ *	
137	0000	.	.	;*****	
138	000A	.	.	DECRDX EQU 10 ;DECIMAL NUMBERS	
139	0008	.	.	OCTRDX EQU 8 ;OCTAL NUMBERS	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	5
141	0000	.	*****		
142	0000	.	; COMMON VARIABLES *		
143	0000	.	*****		
144	9165	.	INTVEC EQU FSTRAM+1450 ;CENTRAL INTERRUPT VECTOR		
145	9168	.	SCNVEC EQU INTVEC+3 ;FOREIGN TERMINAL DISPLY SCA		
146	0000	.	;		
147	FFFF	.	COMMON EQU 1777770 ;UPPER LIMIT OF COMMON AREA		
148	00FF	.	CMBASE EQU COMMON/256 ;MSB OF COMMON ADDRESSES		
149	FF00	.	CMSTOR EQU CMBASE*256 ;MSB ADJUSTMENT FACTOR		
150	0000	.	;		
151	FFFE	.	DISPST EQU COMMON-1 ;DISPLAY REFRESH START PTR		
152	FFFD	.	TRMTYP EQU DISPST-1 ;TERMINAL TYPE NUMBER		
153	FFFC	.	KBDCSW EQU TRMTYP-1 ;KEYBOARD DATACOM SWITCHES		
154	FFFB	.	KBJMPR EQU KBDCSW-1 ;KEYBOARD STRAPS		
155	FFFA	.	KBJMP2 EQU KBJMPR-1 ;SET 2		
156	FFF9	.	KBJMP3 EQU KBJMP2-1 ;SET 3		
157	FFF8	.	CMFLGS EQU KBJMP3-1 ;COMMON FLAGS		
158	FFF7	.	ERRFLG EQU CMFLGS-1 ;ERROR FLAGS		
159	FFF6	.	INTFLG EQU ERRFLG-1 ;INTERRUPT FLAG		
160	FFF5	.	PRCCTL EQU INTFLG-1 ;PROCESSOR CONTROL FLAGS		
161	FFF4	.	MDFLG1 EQU PRCCTL-1 ;TERMINAL MODE FLAGS 1		
162	FFF3	.	MDFLG2 EQU MDFLG1-1 ;AND 2		
163	FFF1	.	MSGPT1 EQU MDFLG2-2 ;MESSAGE POINTERS		
164	FFEF	.	MSGPT2 EQU MSGPT1-2 ;.		
165	FFED	.	MSGPT3 EQU MSGPT2-2 ;.		
166	FFEB	.	MSGPT4 EQU MSGPT3-2 ;.		
167	FFE9	.	MSGPT5 EQU MSGPT4-2 ;.		
168	FFE7	.	MSGPT6 EQU MSGPT5-2 ;.		
169	FFE5	.	MSGPT7 EQU MSGPT6-2 ;.		
170	FFE3	.	MSGPT8 EQU MSGPT7-2 ;.		
171	FFE1	.	CTIVEC EQU MSGPT8-2 ;CTU INTERRUPT VECTOR		
172	FFE0	.	CTIJMP EQU CTIVEC-1 ;JUMP CODE FOR VECTOR		
173	FFDE	.	IODATA EQU CTIJMP-2 ;ESQ SEQ PARM ACCUMULATOR		
174	FFDD	.	IOCSGN EQU IODATA-1 ;SIGN FOR PARAMETER		
175	FFDC	.	IOPSGN EQU IOCSGN-1 ;PARAMETER SIGN		
176	FFDB	.	PARM1 EQU IOPSGN-1 ;ESCAPE SEQUENCE PARAMETERS		
177	FFDA	.	PARM2 EQU PARM1-1 ;.		
178	FFD9	.	PARM3 EQU PARM2-1 ;.		
179	FFD8	.	PARM4 EQU PARM3-1 ;.		
180	FFD7	.	PARM5 EQU PARM4-1 ;.		
181	FFD5	.	PARM6 EQU PARM5-2 ;.		
182	FFD4	.	RADIX EQU PARM6-1 ;RADIX OF PARAMETERS		
183	FFD2	.	RNGTA EQU RADIX-2 ;CHAR FUNCTION TABLE ADDRESS		
184	FFD1	.	ESCFLG EQU RNGTA-1 ;ESCAPE SEQUENCE FLAG		
185	0000	.	;		
186	0000	.	;		
187	FFD0	.	RSTTMR EQU ESCFLG-1 ;SOFT RESET TIMER		
188	0000	.	; * * * * *		
189	0000	.	; END OF COMMON EQUATES		
190	0000	.	*****		

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE   6
=====
192      0000      . . .      ;*****
193      0000      . . .      ; KEYBOARD ENTRY VECTOR POINTERS *
194      0000      . . .      ;*****
195      4800      . . .      ZKBBAS EQU 44000Q      ;KEYBOARD START ADDRESS
196      4802      . . .      ZINIKB EQU ZKBBAS+2    ;INITIALIZE KEYBOARD
197      4805      . . .      ZGETKY EQU ZINIKB+3    ;GET KEYBOARD KEY
198      4808      . . .      ZKBCTL EQU ZGETKY+3    ;PERFORM KEYBOARD CONTRL
199      480B      . . .      ZKBMON EQU ZKBCTL+3    ;MONITOR KEYBOARD
200      480E      . . .      ZSTMD1 EQU ZKBMON+3    ;SET MODE 1 FLAGS
201      4811      . . .      ZCLMD1 EQU ZSTMD1+3    ;CLEAR MODE 1 FLAGS
202      4814      . . .      ZBELL EQU ZCLMD1+3     ;SOUND THE BELL
203      4817      . . .      ZSTXMT EQU ZBELL+3     ;TURN ON TRANSMIT LED
204      481A      . . .      ZCLXMT EQU ZSTXMT+3    ;TURN OFF TRANSMIT LED
205      481D      . . .      ZSTJPR EQU ZCLXMT+3    ;SET JUMPERS ESC SEQ ROUTINE
206      4820      . . .      ZSTLKY EQU ZSTJPR+3    ;SET LATCHING KEYS ROUTINE
207      4823      . . .      ZALPCK EQU ZSTLKY+3    ;ALPHA KEY ENTRY CHECK
208      4826      . . .      ZNUMCK EQU ZALPCK+3    ;NUMERIC KEY ENTRY CHECK
209      0000      . . .      ;
210      0000      . . .      ; KEYBOARD CONSTANTS
211      0000      . . .      ;
212      4829      . . .      FRSALE EQU ZNUMCK+3    ;INITIAL ALTERNATE CHAR SET
213      482A      . . .      ALTOUT EQU FRSALE+1    ;INITIAL ALTERNATE CHAR OUT
214      0000      . . .      ;
215      0000      . . .      ; KEYBOARD CONTROL CALLS
216      0000      . . .      ;
217      0001      . . .      LOCKKB EQU 1           ;LOCK KEYBOARD
218      0002      . . .      UNLKB EQU 2            ;UNLOCK KEYBOARD
219      0003      . . .      RPTKEY EQU 3           ;REPEAT LAST KEY HIT
220      0004      . . .      STBLMD EQU 4           ;SET PERMANENT BLOCK MODE
221      0005      . . .      STRTST EQU 5           ;START SELF-TEST
222      0006      . . .      ENDTST EQU 6           ;END SELF-TEST
223      0007      . . .      RSETKB EQU 7           ;RESET KEYBOARD
224      0008      . . .      CKIOKY EQU 8           ;CHECK FOR I/O CONTROL KEY
225      0009      . . .      STPRPT EQU 9           ;STOP KEY REPEAT
226      000A      . . .      CKBRKY EQU 10          ;CHECK FOR BREAK KEY DOWN
227      000B      . . .      SWCHAR EQU 11          ;SWITCH CHARACTER SET
228      000C      . . .      SETFRN EQU 12          ;UPDATE FOREIGN MODE
229      000D      . . .      STCHST EQU 13          ;SET FOREIGN OUTPUT MODE
230      000E      . . .      FRNMD1 EQU 14          ;SET FOREIGN MODE 1
231      000F      . . .      FRNMD2 EQU 15          ;SET FOREIGN MODE 2
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
233	0000	. . .	;*****	
234	0000	. . .	;	
235	0000	. . .	;	
236	0000	. . .	;	
237	0000	. . .	;*****	
238	5000	. . .	ZDCBAS EQU 500000 ;DATACOM START ADDRESS	
239	5002	. . .	TRIGGR EQU ZDCBAS+2 ;BLOCK TRANSFER TRIGGER	
240	5003	. . .	RECSEP EQU TRIGGR+1 ;RECORD SEPARATOR CHARACTER	
241	5004	. . .	BLKTRM EQU RECSEP+1 ;BLOCK TERMINATOR CHARACTER	
242	5005	. . .	DCJMSK EQU BLKTRM+1 ;DATA COMM JUMPER MASK	
243	5006	. . .	DCJMS2 EQU DCJMSK+1 ;DATA COMM JUMPER MASK #2	
244	0000	. . .	;*****	
245	0000	. . .	;	
246	0000	. . .	;	
247	0000	. . .	;	
248	0000	. . .	;*****	
249	5008	. . .	ZINIDC EQU ZDCBAS+100 ;INITIALIZE DATACOM	
250	500B	. . .	ZIN2DC EQU ZINIDC+3 ;INITIALIZATION CONTINUATOR	
251	500E	. . .	ZDCMON EQU ZIN2DC+3 ;MONITORING ROUTINE	
252	5011	. . .	ZDCCTL EQU ZDCMON+3 ;MISC CONTROL FUNCTIONS	
253	5014	. . .	ZDCTST EQU ZDCCTL+3 ;SELF-TEST	
254	5017	. . .	ZGETDC EQU ZDCTST+3 ;GET DC CHARACTER	
255	501A	. . .	ZPUTDC EQU ZGETDC+3 ;PUT DC CHARACTER	
256	501D	. . .	ZGTBIN EQU ZPUTDC+3 ;GET BINARY DC CHARACTER	
257	5020	. . .	ZSTBIN EQU ZGTBIN+3 ;START BINARY OUTPUT	
258	5023	. . .	ZNDBIN EQU ZSTBIN+3 ;END BINARY OUTPUT	
259	5026	. . .	ZDCINT EQU ZNDBIN+3 ;DATACOM INTERRUPTS	
260	0000	. . .	;*****	
261	0000	. . .	;	
262	0000	. . .	;	
263	0000	. . .	;	
264	0000	. . .	;*****	
265	0000	. . .	CLRTRG EQU 0 ;CLEAR BLOCK TRANSFER TRIGGER	
266	0001	. . .	SETTRG EQU 1 ;SET BLOCK TRANSFER TRIGGER	
267	0002	. . .	RSETDC EQU 2 ;RESET DATACOM	
268	0003	. . .	SETREM EQU 3 ;SET REMOTE MODE	
269	0004	. . .	SETLCL EQU 4 ;SET LOCAL MODE	
270	0005	. . .	PUTBRK EQU 5 ;OUTPUT BREAK SIGNAL	
271	0006	. . .	DISCNT EQU 6 ;MODEM DISCONNECT	
272	0007	. . .	ENDBLK EQU 7 ;TERMINATE OUTPUT MESSAGE	
273	0008	. . .	SFTMON EQU 8 ;ENTER MONITOR MODE	
274	0009	. . .	SETNRM EQU 9 ;ENTER NORMAL MODE	
275	000A	. . .	FSTBIN EQU 10 ;ENTER FAST BINARY OUT MODE	
276	000B	. . .	SNDATN EQU 11 ;SEND ATTENTION CODE	
277	000C	. . .	SNDFCT EQU 12 ;SEND FUNCTION DATA	
278	000D	. . .	PROMPT EQU 13 ;SEND PROMPT CODE	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE   8
=====
280      0000      . . .      ;*****
281      0000      . . .      ; ALTERNATE I/O ENTRY VECTORS *
282      0000      . . .      ;*****
283      9200      . . .      ALTORG EQU 1110000 ;ALTERNATE IO START (36.5K)
284      9202      . . .      ZINIAL EQU ALTORG+2 ;INITIALIZATION ROUTINE
285      9205      . . .      ZIN2AL EQU ZINIAL+3 ;INITIALIZATION CONTINUATOR
286      9208      . . .      ZINTAL EQU ZIN2AL+3 ;INTERRUPT PROCESSOR
287      920B      . . .      ZMONAL EQU ZINTAL+3 ;MONITORING ROUTINE
288      920E      . . .      ZGETAL EQU ZMONAL+3 ;INPUT ROUTINE
289      9211      . . .      ZPUTAL EQU ZGETAL+3 ;OUTPUT ROUTINE
290      9214      . . .      ZCTLAL EQU ZPUTAL+3 ;CONTROL ROUTINE
291      9217      . . .      ZSTAAL EQU ZCTLAL+3 ;STATUS ROUTINE
292      921A      . . .      ZMSGAL EQU ZSTAAL+3 ;ALTERNATE DEVICE NAME
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE   9
=====
294      0000      . . .      ;*****
295      0000      . . .      ; GRAPHICS ENTRY VECTORS AND SYMBOLS_
296      0000      . . .      ;*****
297      0000      . . .      ;
298      6000      . . .      ZGBASE EQU 600009 ;START GRAPHICS AT 24K
299      6002      . . .      ZINGR EQU ZGBASE+2 ;INIT. GRAPHICS--HARD RESET
300      6005      . . .      ZGSOFT EQU ZINGR+3 ;SOFT RESET
301      6008      . . .      ZGSTUP EQU ZGSOFT+3 ;ESC*--SET UP FOR GRAPHICS
302      600B      . . .      ZANCHK EQU ZGSTUP+3 ;A/N CURSOR STORE CHECK
303      600E      . . .      ZGCKEY EQU ZANCHK+3 ;G CURSOR KEY SCAN
304      6011      . . .      ZRELGC EQU ZGCKEY+3 ;G CURSOR KEY RELEASED
305      6014      . . .      ZTINTR EQU ZRELGC+3 ;TIMER INTERRUPT
306      6017      . . .      ZVR EQU ZTINTR+3 ;VERTICAL RETRACE
307      601A      . . .      ZAPMOF EQU ZVR+3 ;TURN AUTO PLOT MENU OFF
308      601D      . . .      ZAPSCN EQU ZAPMOF+3 ;AP VALUE SCAN
309      6020      . . .      ZCR EQU ZAPSCN+3 ;CARRIAGE RETURN
310      6023      . . .      ZMUCHK EQU ZCR+3 ;SEE IF AP MENU ON
311      6026      . . .      ZINFIX EQU ZMUCHK+3 ;INSERTED CHARACTER KLUGE
312      6029      . . .      ZAPCHK EQU ZINFIX+3 ;AP KEYBOARD ENTRY
313      602C      . . .      ZGFUNC EQU ZAPCHK+3 ;GRAPHICS KEYPAD FUNCTION
314      602F      . . .      ZTKSUP EQU ZGFUNC+3 ;GS--SET FOR TEK VECTORS
315      6032      . . .      ZPAGE EQU ZTKSUP+3 ;ESC FF--DU TEK 'PAGE'
316      6035      . . .      ZSTGIN EQU ZPAGE+3 ;ESC SUB--START TEK GIN MODE
317      6038      . . .      ZTKHC EQU ZSTGIN+3 ;ESC ETB--MAKE TEK HARDCOPY
318      603B      . . .      ZTKCUR EQU ZTKHC+3 ;ESC ENQ--READ CURSOR POSITI
319      603E      . . .      ZTKCLR EQU ZTKCUR+3 ;CLEAR ECHOPLEX SUPPRESS
320      6041      . . .      ZHT EQU ZTKCLR+3 ;PROCESS HT
321      6044      . . .      ZVT EQU ZHT+3 ;PROCESS VT
322      6047      . . .      ZBS EQU ZVT+3 ;PROCESS BS
323      604A      . . .      ZLF EQU ZBS+3 ;PROCESS LF
324      604D      . . .      ZDPTST EQU ZLF+3 ;TEST FOR A/N OR G DISPLAY
325      6050      . . .      ZVID1 EQU ZDPTST+3 ;SUPPRESS GRAPHICS, ALLOW A/N
326      6053      . . .      ZVID2 EQU ZVID1+3 ;RESTORE STATE OF GRAFIX,A/N
327      6056      . . .      ZAPCR EQU ZVID2+3 ;AUTO PLOT CARRIAGE RETURN
328      6059      . . .      ZTKSTR EQU ZAPCR+3 ;INITIALIZE TEK STRAPS
329      605C      . . .      ZMUTB EQU ZTKSTR+3 ;ADDRESS OF MENU TABLE
330      605E      . . .      ZGSTAT EQU ZMUTB+2 ;SEND GRAPHICS STATUS
331      6061      . . .      ZGRTST EQU ZGSTAT+3 ;TEST FOR GRAPHICS GET
332      6064      . . .      ZGGINT EQU ZGRTST+3 ;INITIALIZE FOR GRAPHICS GET
333      6067      . . .      ZGRGET EQU ZGGINT+3 ;GET GRAPHICS DATA
334      606A      . . .      ZAPHME EQU ZGRGET+3 ;HOME AUTO PLOT CURSOR
335      606D      . . .      ZGTEST EQU ZAPHME+3 ;GRAPHICS SELF TEST
336      6070      . . .      ZCHKTK EQU ZGTEST+3 ;SEE IF IN TEK MODE
337      6073      . . .      ZANCUR EQU ZCHKTK+3 ;MOVE GC WITH A/N KEYS
338      6076      . . .      ZAPLF EQU ZANCUR+3 ;AUTO PLOT LINE FEED
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 10
=====
340      0000      . . .      ;*****
341      0000      . . .      ; GRAPHICS SYMBOLS
342      0000      . . .      ;*****
343      0020      . . .      SUPCHR EQU 400          ;ECHOPLEX SUPPRESS (TEK)
344      0020      . . .      AVINHB EQU 400         ;A/N VIDEO IS INHIBITED
345      0040      . . .      ACINHB EQU 1000        ;A/N CURSOR IS INHIBITED
346      0098      . . .      GFUNMX EQU 2300        ;MAX GRAPHICS FUNCTION CODE
347      0010      . . .      GINMOD EQU 200         ;IN TEK GIN MODE
348      0007      . . .      BEL      EQU 70         ;BELL CONTROL CODE
349      001F      . . .      US       EQU 370        ;UNIT SEPARATOR
350      90AD      . . .      ZTKFLG  EQU 110255Q    ;TEK MODE FLAGS
351      90B2      . . .      ZGFLG1  EQU 110262Q    ;GFLGS1
352      906B      . . .      ZGSBLK  EQU 110153Q    ;GRAPHICS STATUS BLOCK #
353      FB96      . . .      ZAPFLG  EQU 175626Q    ;AUTO PLOT FLAGS
354      9097      . . .      ZGFLG6  EQU 110227Q    ;GRAPHICS TEXT FLAGS
355      0002      . . .      APIP    EQU 20         ;AUTO PLOT IN PROGRESS
356      0002      . . .      GTEXT   EQU 20         ;GRAPHICS TEXT UN
357      0080      . . .      LABEL   EQU 200Q       ;SINGLE RECORD LABEL
358      0004      . . .      HP2648  EQU 40          ;STATUS BIT FOR TRMTYP
359      0200      . . .      MINMEM  EQU 1000Q      ;MINIMUM DISPLAY MEM NEEDED
360      0000      . . .      RIGHT   EQU 0          ;GRAPHICS CURSOR RIGHT
361      0001      . . .      DOWN    EQU 1
362      0002      . . .      LEFT    EQU 2
363      0003      . . .      UP      EQU 3
=====
```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 11
365	0000	.	*****	
366	0000	.	; ASCII CHARACTER EQUATES *	
367	0000	.	*****	
368	0000	.	NULL EQU 00 ;NULL	
369	000A	.	LF EQU 120 ;LINE FEED	
370	000C	.	FF EQU 140 ;FORM FEED	
371	000D	.	CR EQU 150 ;RETURN	
372	000E	.	SO EQU 0160	
373	000F	.	SI EQU 0170	
374	0012	.	DC2 EQU 220 ;DEVICE CONTROL 2	
375	0013	.	DC3 EQU 230 ;DEVICE CONTROL 3	
376	0018	.	ESC EQU 330 ;ESCAPE	
377	0020	.	CTLLIM EQU 400 ;CONTROL CODE UPPER LIMIT	
378	0020	.	ABLNK EQU 0400 ;ASCII BLANK	
379	0026	.	AMPSND EQU 460 ;(&) - AMPERSAND	
380	0027	.	QUOTE EQU 470 ;(') - SINGLE QUOTE	
381	0029	.	ARPARN EQU 510 ;(]) - RIGHT PARENTHESIS	
382	002B	.	PLUS EQU 530 ;PLUS SIGN	
383	002C	.	COMMA EQU 540 ;COMMA	
384	002D	.	MINUS EQU 550 ;MINUS SIGN	
385	002E	.	PERIOD EQU 560 ;(.) - PERIOD	
386	002F	.	SLANT EQU 570 ;(/) - SLANT	
387	0030	.	ZERO EQU 600 ;ASCII ZERO	
388	0032	.	TWO EQU 620 ;ASCII TWO	
389	0033	.	THREE EQU 630 ;ASCII THREE	
390	0034	.	FOUR EQU 640 ;ASCII FOUR	
391	0035	.	FIVE EQU 650 ;ASCII FIVE	
392	0036	.	SIX EQU 660 ;ASCII SIX	
393	0037	.	SEVEN EQU 670 ;ASCII SEVEN	
394	0000	.	;	
395	0040	.	ATSIGN EQU 1000 ;"AT" SIGN (@)	
396	0041	.	A EQU 1010 ;UPPER CASE A	
397	0043	.	C EQU 1030 ;UPPER CASE C	
398	0044	.	D EQU 1040 ;UPPER CASE D	
399	0046	.	F EQU 1060 ;UPPER CASE F	
400	0048	.	H EQU 1100 ;UPPER CASE H	
401	004C	.	L EQU 1140 ;UPPER CASE L	
402	004E	.	N EQU 1160 ;UPPER CASE N	
403	0050	.	P EQU 1200 ;UPPER CASE P	
404	0052	.	R EQU 1220 ;UPPER CASE R	
405	0053	.	S EQU 1230 ;UPPER CASE S	
406	0054	.	T EQU 1240 ;UPPER CASE T	
407	0055	.	U EQU 1250 ;UPPER CASE U	
408	0059	.	Y EQU 1310 ;UPPER CASE Y	
409	005A	.	Z EQU 1320 ;UPPER CASE Z	
410	005B	.	LFTBKT EQU 1330 ;LEFT BRACKET	
411	005C	.	ABCKSL EQU 1340 ;(\) - BACK SLANT	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  12
=====
413      0000      . . .      ;*****
414      0000      . . .      ; LOWER CASE EQUATES *
415      0000      . . .      ;*****
416      0061      . . .      SMALLA EQU 141Q      ;LOWER CASE A
417      0063      . . .      ALCC EQU 143Q      ;ASCII LOWER CASE C
418      0064      . . .      SMALLD EQU 144Q      ;LOWER CASE D
419      0066      . . .      SMALLF EQU 146Q      ;LOWER CASE F
420      0069      . . .      SMALLI EQU 151Q      ;LOWER CASE I
421      0068      . . .      SMALLK EQU 153Q      ;LOWER CASE K
422      0070      . . .      SMALLP EQU 160Q      ;LOWER CASE P
423      0078      . . .      SMALLX EQU 170Q      ;LOWER CASE X
424      007B      . . .      LFTBRC EQU 173Q      ;LEFT BRACE
425      007C      . . .      VRTBAR EQU 174Q      ;VERTICAL BAR
426      007F      . . .      ADEL EQU 177Q      ;DELETE (RUBOUT)
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 13
428	0000	. . .	;*****	
429	0000	. . .	; DISPLAY FLAGS EQUATES *	
430	0000	. . .	;*****	
431	00BF	. . .	ENHLIM EQU 277Q ;MAXIMUM ENHANCEMENT CODE	
432	00C0	. . .	STPR EQU 300Q ;START PROTECTED FIELD	
433	00C1	. . .	ENDPR EQU 301Q ;END PROTECTED FIELD	
434	00C2	. . .	XMONLY EQU 302Q ;START TRANSMIT-ONLY FIELD	
435	00C3	. . .	FILL EQU 303Q ;EOL FILL CHARACTER	
436	00C4	. . .	STPFLG EQU 304Q ;NON-DISPLAYING TERMINATOR	
437	00C5	. . .	ALPHA EQU 305Q ;ALPHABETIC ONLY	
438	00C6	. . .	NUMBER EQU 306Q ;NUMERIC ONLY	
439	00C7	. . .	ALPHNM EQU 307Q ;ALPHANUMERIC FIELD	
440	00C8	. . .	SFKYAT EQU 310Q ;SOFT KEY ATTRIBUTE FIELD	
441	0000	. . .	;	
442	00C4	. . .	FLDSEP EQU 304Q ;FIELD SEPARATOR FOR I/O BUF	
443	00CC	. . .	EOL EQU 314Q	
444	00CE	. . .	EOP EQU 316Q	
445	00D0	. . .	LNKLIM EQU 320Q ;LOWEST VALUE FOR A LINK	
446	0800	. . .	NUM2K EQU 4000Q ;NUMBER 2048 (2K)	
447	8000	. . .	B15 EQU 100000Q ;BIT 15	
448	00C3	. . .	JMP EQU 303Q ;JUMP INSTRUCTION CODE	
449	00C9	. . .	RET EQU 311Q ;RETURN INSTRUCTION CODE	
450	0000	. . .	;*****	
451	0000	. . .	; MISCELLANEOUS EQUATES *	
452	0000	. . .	;*****	
453	0017	. . .	MAXROW EQU 23 ;MAXIMUM ROW NUMBER	
454	004F	. . .	MAXCOL EQU 79 ;MAXIMUM COLUMN NUMBER	
455	0010	. . .	SFTEND EQU 16 ;LAST SOFT KEY DEFINITION RO	
456	0008	. . .	BELLIM EQU 8 ;SPACE FROM RHTMGN FOR BELL	
457	000F	. . .	BLKSM EQU 17Q ;BLOCK SIZE MASK	
458	0010	. . .	BLKSZ EQU 16 ;BLOCK SIZE	
459	0020	. . .	IOERRB EQU 40Q ;I/O ERROR STATUS BIT	
460	0001	. . .	REXMIT EQU 1Q ;RE-TRANSMIT I/O FLAG	
461	0002	. . .	BINXMT EQU 2 ;SEND BINARY DATA	
462	0032	. . .	SFTDLY EQU 50 ;SOFT RESET PERIOD - .50 SEC	
463	0080	. . .	NOSIGN EQU 200Q ;NO SIGN FLAG FOR INPUT DATA	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  14
=====
 465      0000      . . .      ;*****
 466      0000      . . .      ; I/O MODULE EQUATES *
 467      0000      . . .      ;*****
 468      0000      . . .      RESET EQU 00          ;RESET TERMINAL VECTOR
 469      0001      . . .      RSTJMP EQU 10         ;VECTOR FOR RESTART "PCHL"
 470      0070      . . .      PROCSR EQU 1600       ;PROCESSOR "OUT" PORT
 471      0080      . . .      IOBASE EQU 2000       ;I/O ADDRESS MSB'S
 472      0000      . . .      ;
 473      0000      . . .      ;  KEYBOARD
 474      0000      . . .      ;
 475      8300      . . .      IOKB EQU 30+IOBASE*256 ;MODULE 11 BASE ADDRESS
 476      8380      . . .      IOKBCU EQU IOKB+200Q  ;RESET KEY CONTROL
 477      0002      . . .      RSTON EQU 20          ;RESET ON
 478      0004      . . .      RSTOFF EQU 40         ;RESET OFF
 479      0000      . . .      ;***** GRAPHICS MUDIFICATION *****
 480      0009      . . .      NMFCTK EQU 9          ;NUMBER OF FUNCTION KEYS
 481      00EF      . . .      SFTCR EQU 3570        ;SOFT RETURN KEY CODE
 482      0000      . . .      ;*****
 483      0000      . . .      ;
 484      0000      . . .      ;  CURSOR CONTROL
 485      0000      . . .      ;
 486      8700      . . .      IODISP EQU 70+IOBASE*256 ;MODULE 13 BASE ADDRESS
 487      8700      . . .      IOCRCL EQU IODISP+0   ;CURSOR COLUMN ADDRESS
 488      8720      . . .      IOCRRW EQU IODISP+40Q ;CURSOR ROW ADDRESS
 489      0020      . . .      MAYEOP EQU 400        ;DMA ON, EOP IF DMA ROW = RO
 490      0040      . . .      MAYEOL EQU 100Q       ;DMA OFF, SKIP EOP IF ROWS =
 491      0060      . . .      DMAOFF EQU 140Q       ;DMA OFF
 492      0080      . . .      CRTOFF EQU 200Q       ;DISPLAY OFF
 493      0082      . . .      INVRS EQU 202Q        ;INVERSE VIDEO ON
 494      0080      . . .      NORMAL EQU 200Q       ;NORMAL VIDEO ON
 495      0000      . . .      ;
 496      0000      . . .      ;  CARTRIDGE TAPE
 497      0000      . . .      ;
 498      8B00      . . .      IOCTU EQU 130+IOBASE*256 ;MODULE 15 BASE ADDRES
 499      8B00      . . .      IOCTCO EQU IOCTU+0Q   ;COMMAND TO CTU
 500      8B00      . . .      IOCTSI EQU IOCTU+0Q   ;STATUS FROM CTU
 501      8B20      . . .      IOCTDO EQU IOCTU+40Q  ;DATA TO CTU
 502      8B20      . . .      IOCTDI EQU IOCTU+40Q  ;DATA FROM CTU
=====

```

=====

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 15
------	-----	--------	------	-------------------	---------

=====

504	0000	.	.	.	;
505	0000	.	.	.	; 9866 PRINTER
506	0000	.	.	.	;
507	8D00	.	.	.	IOPTR1 EQU 150+IOBASE*256 ;MODULE 16 BASE ADDRESS
508	8D20	.	.	.	PTROT1 EQU IOPTR1+400 ;PRINTER DATA OUT
509	8D00	.	.	.	PTRST1 EQU IOPTR1+00 ;PRINTER STATUS IN
510	8D02	.	.	.	PTRCL1 EQU IOPTR1+20 ;PRINTER CLEAR
511	0000	.	.	.	;
512	0000	.	.	.	; RS-232 PRINTER
513	0000	.	.	.	;
514	8500	.	.	.	IOPTR2 EQU 50+IOBASE*256 ;MODULE 12 BASE ADDRESS
515	8540	.	.	.	PTROT2 EQU IOPTR2+1000 ;INTERFACE CONTROL OUT
516	8520	.	.	.	PTRST2 EQU IOPTR2+400 ;PRINTER STATUS IN
517	8560	.	.	.	PTRDA2 EQU IOPTR2+1400 ;PRINTER DATA OUT
518	8540	.	.	.	PTRCF2 EQU IOPTR2+1000 ;OPTION JUMPERS IN

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 16
=====
520      0000      . . .      ;*****
521      0000      . . .      ; PRINTER EQUATES *
522      0000      . . .      ;*****
523      0000      . . .      ;
524      0000      . . .      ; RS-232 OPTION STRAPS
525      0000      . . .      ;
526      0000      . . .      ; BITS 2-0 MEANING IF SET
527      0000      . . .      ; 000 EXT BAUD RATE
528      0000      . . .      ; 001 110 "
529      0000      . . .      ; 010 150 "
530      0000      . . .      ; 011 300 "
531      0000      . . .      ; 100 1200 "
532      0000      . . .      ; 101 2400 "
533      0000      . . .      ; 110 4800 "
534      0000      . . .      ; 111 9600 "
535      0000      . . .      ;
536      0000      . . .      ; BIT 3 PARITY SELECT
537      0000      . . .      ; 1 EVEN
538      0000      . . .      ; 0 ODD
539      0000      . . .      ;
540      0000      . . .      ; BIT 4 PARITY INHIBIT
541      0000      . . .      ; 1 NO PARITY
542      0000      . . .      ; 0 PARITY
543      0000      . . .      ; BITS 7-5 # OF FILLS
544      0000      . . .      ; 000 HANDSHAKE DEVICE
545      0000      . . .      ; 001 8
546      0000      . . .      ; 010 16
547      0000      . . .      ; 011 24
548      0000      . . .      ; 100 32
549      0000      . . .      ; 101 40
550      0000      . . .      ; 110 48
551      0000      . . .      ; 111 56
552      0000      . . .      ;*****
553      0000      . . .      ; DRIVER EQUATES *
554      0000      . . .      ;*****
555      05DC      . . .      PTDLY EQU 1500 ;15 SECOND PRINTER TIME OUT
556      0000      . . .      ;*****
557      0000      . . .      ; 9866 PRINTER EQUATES *
558      0000      . . .      ;*****
559      0001      . . .      PTRDY1 EQU 1 ;PRINTER READY
560      0080      . . .      PTRPO1 EQU 2000 ;PRINTER OUT OF PAPER
561      0000      . . .      ;*****
562      0000      . . .      ; RS-232 PRINTER EQUATES *
563      0000      . . .      ;*****
564      0002      . . .      PTRDY2 EQU 2 ;PRINTER READY MASK
565      0040      . . .      PTRSB2 EQU 1000 ;RS-232 S3 LINE STROBE
566      0020      . . .      PTRDL2 EQU 400 ;PRINTER READY MASK
567      00E0      . . .      PTRHD2 EQU 3400 ;RS-232 HANDSHAKE PROTOCOL
568      001F      . . .      PTRBD2 EQU 370 ;PARITY AND BAUD RATE MASK
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 17
=====
570      0000      . . .      ;*****
571      0000      . . .      ; VARIABLE SPACE ALLOCATION *
572      0000      . . .      ;*****
573      0000      . . .      ;*****
574      0000      . . .      ; MUST BE ON 256 BYTE BOUNDRY
575      F7FF      . . .      DSPLIM EQU 173777Q ;DISPLAY LOWER LIMIT
576      0000      . . .      ;*****
577      00D0      . . .      LWDSP EQU 150000Q/256 ;DISPLAY LOWER LIMIT
578      FC00      . . .      IOBUF EQU 176000Q
579      00FC      . . .      IOBUFH EQU IOBUF/256
580      0000      . . .      IOBUFL EQU -IOBUFH*256+IOBUF
581      FC00      . . .      IOBUF1 EQU 176000Q
582      FD00      . . .      IOBUF2 EQU 176400Q
583      FE4F      . . .      DSPSTR EQU 177000Q+79 ;MESSAGE BUFFER
584      0100      . . .      PTRBLN EQU 256 ;PRINTER INPUT BUFFER SIZE
585      0000      . . .      ;*****
586      0000      . . .      ; OPERATING SYSTEM STORAGE *
587      0000      . . .      ;*****
588      9160      . . .      STACK EQU FSTRAM+140Q ;STACK AREA (96 BYTES)
589      FFD0      . . .      OPSTOR EQU 177720Q ;VARIABLES STORAGE AREA
590      00FF      . . .      BASEH EQU OPSTOR/256 ;MSB OF DATA PAGE ADDRESSE
591      FF00      . . .      BASE EQU BASEH*256 ;DATA PAGE BASE ADDRESS
592      00FE      . . .      BASEH2 EQU BASEH-1 ;BASE VALUES FOR SECOND PAGE
593      FE00      . . .      BASE2 EQU BASEH2*256 ;OF VARIABLES SPACE
594      0000      . . .      ;*****
595      0000      . . .      ; VARIABLE SUBROUTINE CALL *
596      0000      . . .      ;*****
597      FFCD      . . .      ECONF EQU OPSTOR-3 ;JUMP SUBROUTINE
598      FFCE      . . .      CNTFAD EQU ECONF+1 ;CHARACTER FUNCTION ADDRESS
=====
  
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
600	0000	. . .	;*****	18
601	0000	. . .	; NORMAL/SOFT KEY SWAPPED DISPLAY PARAMETERS *	
602	0000	. . .	;*****	
603	FFC8	. . .	TOPLIN EQU ECONF-2 ;LSB PART OF NEXT LINE	
604	0000	. . .	; POINTER IN TOP DISPLAY	
605	0000	. . .	LINE	
606	FFC9	. . .	LSTLIN EQU TOPLIN-2 ;POINTER TO LSB PART OF	
607	0000	. . .	; NEXT LINE POINTER IN	
608	0000	. . .	; LAST LINE PROCESSED	
609	FFC8	. . .	LSTCOL EQU LSTLIN-1 ;COLUMN AND ROW POSITION OF	
610	FFC7	. . .	LSTROW EQU LSTCOL-1 ;LAST CHARACTER PROCESSED	
611	0000	. . .	; (CORRESPONDS TO CHARACTER	
612	0000	. . .	; GIVEN BY "CURADR")	
613	FFC6	. . .	LSTDCD EQU LSTROW-1 ;LAST DISPLAY CODE USED	
614	FFC5	. . .	LSTFMT EQU LSTDCD-1 ;LAST FORMAT CONTROL USED	
615	FFC3	. . .	CURADR EQU LSTFMT-2 ;ADDRESS OF LAST CHARACTER	
616	0000	. . .	; PROCESSED	
617	FFC2	. . .	PROFLD EQU CURADR-1 ;PROTECT STATE OF (CURADR)	
618	0000	. . .	; = -1, PROTECTED	
619	0000	. . .	; # -1, NOT PROTECTED	
620	0000	. . .	;*****	
621	0000	. . .	; CURRENT CURSOR VALUES *	
622	0000	. . .	;*****	
623	FFC1	. . .	CURCOL EQU PROFLD-1 ;CURRENT COLUMN AND ROW	
624	FFC0	. . .	CURROW EQU CURCOL-1 ;POSITION OF CURSOR	
625	FFBF	. . .	LFTMGN EQU CURROW-1 ;LEFT MARGIN SETTING	
626	FFBE	. . .	RHTMGN EQU LFTMGN-1 ;RIGHT MARGIN SETTING	
627	000F	. . .	NUMSWP EQU ECONF-RHTMGN ;# OF SWAP VARIABLES	
628	FFAF	. . .	SWPSTR EQU RHTMGN-NUMSWP ;SWAP BUFFER	
629	FFAE	. . .	DSPTYP EQU SWPSTR-1 ;DISPLAY CURRENTLY ENABLED	
630	0000	. . .	; 0 = NORMAL DISPLAY	
631	0000	. . .	; -1 = SOFT KEY DISPLAY	
632	0000	. . .	;*****	
633	0000	. . .	; FIXED DISPLAY PARAMETERS (NOT SWAPPED) *	
634	0000	. . .	;*****	
635	FFAC	. . .	FRBLKS EQU DSPTYP-2 ;FREE BLOCKS LIST HEAD	
636	FFAA	. . .	DSPBGN EQU FRBLKS-2 ;LOW ADDRESS OF DISPLAY AREA	
637	FFA8	. . .	DSPEND EQU DSPBGN-2 ;HIGH ADDR OF DISPLAY AREA	
638	FFA6	. . .	SFTKYS EQU DSPEND-2 ;SOFT KEY DISPLAY START ADDR	
639	FFA4	. . .	CURFKY EQU SFTKYS-2 ;CURRENT FUNCTION KEY CHAR	
640	FFA3	. . .	TLINO EQU CURFKY-1 ;TOP LINE ABSOLUTE ROW NUMBE	
641	FFA1	. . .	LLINE EQU TLINO-2 ;LAST DISPLAY LINE START ADD	
642	FF9F	. . .	FLINE EQU LLINE-2 ;POINTER TO LSB PART OF NEXT	
643	0000	. . .	; LINE POINTER IN FIRST	
644	0000	. . .	; LINE OF NORMAL DISPLAY	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
646	0000	.	.	;*****	19
647	0000	.	.	; SCRATCH VARIABLES *	
648	0000	.	.	;*****	
649	FF9E	.	.	TEMP1 EQU FLINE-1	
650	FF9D	.	.	TEMP EQU TEMP1-1 ;TEMPORARY STORAGE	
651	FF9C	.	.	CHARIN EQU TEMP-1 ;CHARACTER FROM KEYBOARD	
652	FF9B	.	.	NCHAR EQU CHARIN-1 ;NUMBER OF CHARS TO BE ADDED	
653	FF9A	.	.	NROWS EQU NCHAR-1 ;NO. OF ROWS TO BE ADDED	
654	FF99	.	.	NBLKS EQU NROWS-1 ;NO. OF BLOCKS TO BE ADDED	
655	FF98	.	.	CHSAV EQU NBLKS-1 ;SAVE AREA FOR CHAR	
656	0000	.	.	; PRECEDING LINK	
657	FF96	.	.	LNKSAV EQU CHSAV-2 ;LINK SAVE AREA	
658	FF94	.	.	EOLADR EQU LNKSAV-2 ;ADDR OF LAST EOL	
659	FF92	.	.	FRSTBL EQU EOLADR-2 ;FIRST BLOCK IN DISPL1	
660	FF91	.	.	BLKFIL EQU FRSTBL-1 ;FILL FLAG FOR FNDCHR	
661	FF90	.	.	EOLMV EQU BLKFIL-1 ;FLAG FOR EOLMOV	
662	FF8F	.	.	FILCHR EQU EOLMV-1 ;FILL CHAR SAVE FOR GTBLK	
663	CFFF	.	.	BFSPCE EQU 14777Q ;UPPER LIMIT OF BUFFER	
664	00B0	.	.	LWBUF EQU 130000Q/256 ;LOWER LIMIT	
665	FF8D	.	.	BUFBN EQU FILCHR-2 ;LOW ADDR OF NON-DISPLY BUFF	
666	FF8B	.	.	BUFEND EQU BUFBN-2 ;HIGH ADDR FOR BUFFER	
667	0000	.	.	;*****	
668	0000	.	.	; STORAGE FOR CHARACTERS TO BE STORED *	
669	0000	.	.	;*****	
670	FF8A	.	.	FMTCTL EQU BUFEND-1 ;FORMAT CONTROL TO BE ENTERED	
671	FF89	.	.	DCHAR EQU FMTCTL-1 ;NEXT CHAR TO BE DISPLAYED	
672	FF88	.	.	CHAR EQU DCHAR-1 ;CURRENT CHAR BEING PROCESSED	
673	FF86	.	.	CHKRTN EQU CHAR-2 ;CURRENT TYPE CHECK ROUTINE	
674	FF85	.	.	TMPCOL EQU CHKRTN-1 ;COLUMN # STORAGE FOR RCADDR	
675	0000	.	.	;*****	
676	0000	.	.	; STORAGE FOR CURSOR POSITIONING *	
677	0000	.	.	;*****	
678	FF84	.	.	COUNT EQU TMPCOL-1 ;NUMBER OF BYTES TO FILL	
679	FF83	.	.	NMROLL EQU COUNT-1 ;NUMBER OF LINES TO ROLL	
680	FF82	.	.	ROLLCT EQU NMROLL-1 ;ROLL COUNTER	
681	0000	.	.	;*****	
682	FFDB	.	.	NEWCOL EQU PARM1 ;NEW COLUMN NUMBER	
683	FFDA	.	.	NEWROW EQU PARM2 ;NEW ABSOLUTE ROW NUMBER	
684	FFD9	.	.	SCRNRW EQU PARM3 ;NEW SCREEN ROW SETTING	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  20
=====
686      0000      . . .      ;*****
687      0000      . . .      ; HORIZONTAL TAB TABLE *
688      0000      . . .      ;*****
689      000A      . . .      HTBLLEN EQU 10 ;TABLE LENGTH (= 10 x 8)
690      FF78      . . .      HTBTBL EQU ROLLCT-HTBLLEN
691      0000      . . .      ;*****
692      0000      . . .      ; DISPLAY SEND STORAGE *
693      0000      . . .      ;*****
694      FF77      . . .      CDSPEN EQU HTBTBL-1 ;CURRENT ENHANCEMENT IN
695      FF76      . . .      ENHOUT EQU CDSPEN-1 ;LAST ENHANCEMENT OUT
696      FF75      . . .      CALTST EQU ENHOUT-1 ;CURRENT ALTERNATE SET OUT
697      FF73      . . .      GETADR EQU CALTST-2 ;CURRENT CHARACTER ADDRESS
698      0000      . . .      ;*****
699      0000      . . .      ; FLAGS AND TABLE POINTERS *
700      0000      . . .      ;*****
701      FF72      . . .      CHRSET EQU GETADR-1 ;CURRENT ALTERNATE CHAR SET
702      FF71      . . .      KBFCTK EQU CHRSET-1 ;KEYBOARD FUNCTION CODE
703      0000      . . .      ;*****
704      FF70      . . .      MFLGS EQU KBFCTK-1 ;BLOCK TRANSFER PENDING FLAG
705      0000      . . .      ;*****
706      0100      . . .      SDC2 EQU 10*256 ;DC2 PENDING
707      0200      . . .      SSTAT EQU 20*256 ;TERMINAL STATUS PENDING
708      0400      . . .      SSTAT2 EQU 40*256 ;TERMINAL STATUS 2 PENDING
709      0800      . . .      SDVST EQU 100*256 ;DEVICE STATUS PENDING
710      1000      . . .      SCRSEN EQU 200*256 ;CURSOR SENSE PENDING
711      2000      . . .      SFCTKY EQU 400*256 ;FUNCTION KEY PENDING
712      4000      . . .      SENTER EQU 1000*256 ;DISPLAY SEND PENDING
713      8000      . . .      SDVDUN EQU 2000*256 ;DEVICE DONE PENDING
714      0000      . . .      ;*****
715      FF6F      . . .      MFLGS2 EQU MFLGS-1 ;MAIN CODE MODE FLAGS
716      0000      . . .      ;*****
717      0001      . . .      SDVREC EQU 10 ;DEVICE RECORD PENDING
718      0002      . . .      SBINRY EQU 20 ;BINARY RECORD PENDING
719      0004      . . .      RELSNS EQU 40 ;RELATIVE CURSOR SENSE
720      0008      . . .      ESCINP EQU 100 ;ESC RECEIVED IN BLOCK MODE
721      0010      . . .      FRSCUT EQU 200 ;FIRST SUFT KEY DATA OUT
722      0020      . . .      WRPDEL EQU 400 ;DELETE CHAR W/ WRAP AROUND
723      0040      . . .      WRPFLG EQU 1000 ;LINE WRAP AROUND OCCURRED
724      0080      . . .      NWRWST EQU 2000 ;NEW ABSOLUTE ROW SET
725      0000      . . .      ;*****
726      FF6E      . . .      DFLGS EQU MFLGS2-1 ;DATA TRANSFER FLAGS
727      0000      . . .      ;*****
728      0001      . . .      SDACOM EQU 0010 ;DATACOM/KEYBOARD
729      0002      . . .      CNTXFR EQU 20 ;CONTINUE BUFFER TO DATA COM
730      0004      . . .      NOSEND EQU 40 ;NO DISPLAY DATA TO SEND
731      0008      . . .      SKPTRM EQU 100 ;SKIP BLOCK TERMINATOR
732      0010      . . .      FCTK20 EQU 200 ;FUNCTION KEY TO DISPLAY
733      0040      . . .      KBDLOK EQU 1000 ;KB LOCKED BY ESCAPE SEQUENC
734      0080      . . .      XBF2DS EQU 2000 ;I/O BUFFER TO DISPLAY MODE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 21
736	0000	.	.	*****	
737	FF6D	.	.	TRMFACT EQU DFLGS-1 ;NON-DISPLAYING TERMINATOR	
738	0000	.	.	*****	
739	FFFF	.	.	STPXFR EQU -1 ;TERMINATE TRANSFER	
740	0000	.	.	DELTRM EQU 0 ;DELETE TERMINATOR	
741	0001	.	.	IGNTRM EQU 1 ;IGNORE TERMINATOR	
742	0000	.	.	*****	
743	FF6C	.	.	SPOWL EQU TRMFACT-1 ;SPACE OVERWRITE LATCH	
744	0000	.	.	*****	
745	0020	.	.	SPOWON EQU 400 ;SPOW LATCH ON	
746	00FF	.	.	SPOWOF EQU 3770 ;SPOW LATCH OFF	
747	0000	.	.	*****	
748	FF6B	.	.	MLKROW EQU SPOWL-1 ;MEMORY LOCK ROW	
749	FF6A	.	.	MLKFLG EQU MLKROW-1 ;MEMORY LOCK FLAG	
750	FF69	.	.	LCHAR EQU MLKFLG-1 ;LAST CHARACTER PROCESSED	
751	FF68	.	.	TCHAR EQU LCHAR-1 ;CURRENT TEST PATTERN CHAR	
752	FF67	.	.	CRAFLG EQU TCHAR-1 ;CURSOR ADVANCE FLAG	
753	0000	.	.	*****	
754	0000	.	.	; POINTERS FOR BINARY LOADER *	
755	0000	.	.	*****	
756	FFD5	.	.	LADDR EQU PARM6 ;BYTE ADDRESS PARAMETER	
757	FFDE	.	.	LDATA EQU IODATA ;INPUT DATA ACCUMULATOR	
758	FFD7	.	.	LCHKSM EQU PARM5 ;16-BIT CHECKSUM	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 23
804	0000	.	*****	
805	FF63	.	UNIT0 EQU IOFLG2-1 ;UNIT STATUS	
806	0000	.	*****	
807	0001	.	LPM EQU 10 ;TAPE AT OR BEFORE LOAD POIN	
808	0002	.	LSTFWD EQU 20 ;TAPE LAST MOVED FORWARD	
809	0004	.	FPS EQU 40 ;TAPE WRITE PROTECTED	
810	0008	.	CMDEXC EQU 100 ;SUCCESSFUL COMMAND EXECUTIO	
811	0010	.	DBLHOL EQU 200 ;DOUBLE HOLE FOUND	
812	0020	.	BOT EQU 400 ;TAPE PAST BOT HOLES	
813	0040	.	LP EQU 1000 ;TAPE PAST LP HOLE	
814	0080	.	EW EQU 2000 ;TAPE PAST EW HOLE	
815	0000	.	*****	
816	FF62	.	CNTRLO EQU UNIT0-1 ;DATA TRANSFER FLAGS: *	
817	0000	.	*****	
818	0001	.	EOF EQU 10 ;END OF FILE	
819	0002	.	EVD EQU 20 ;END OF VALID DATA	
820	0004	.	HRDERR EQU 40 ;HARD ERROR	
821	0008	.	SFTERR EQU 100 ;SOFT ERROR	
822	0010	.	HRDER1 EQU 200 ;INTERRUPT ERROR FLAG	
823	0020	.	WRERR EQU 400 ;WRITE ERROR	
824	0040	.	DATATR EQU 1000 ;DATA RECORDED	
825	0000	.	*****	
826	FF61	.	RELTAK EQU CNTRLO-1 ;GAP LENGTH COUNTER	
827	0000	.	*****	
828	FF5F	.	ABSTAK EQU RELTAK-2 ;ABSOLUTE TACH COUNTER	
829	0000	.	*****	
830	405F	.	STRTAK EQU 401370 ;STARTING VALUE	
831	0000	.	*****	
832	FF5E	.	FILNUM EQU ABSTAK-1 ;CURRENT FILE NUMBER	
833	FF5D	.	SFTCNT EQU FILNUM-1 ;SOFT ERRORS PER PASS	
834	FF56	.	OTHER EQU SFTCNT-7 ;STORAGE FOR UNIT NOT SEL.	
835	0000	.	*****	
836	FF55	.	CMND EQU OTHER-1 ;CURRENT CTU COMMAND: *	
837	0000	.	*****	
838	0001	.	RUN EQU 10 ;MOVE TAPE	
839	0002	.	FWD EQU 20 ;FORWARD	
840	0004	.	FST EQU 40 ;FAST	
841	0008	.	REC EQU 100 ;RECORD	
842	0010	.	USL EQU 200 ;SELECT LEFT UNIT	
843	0020	.	GEN EQU 400 ;GAP GENERATE	
844	0040	.	ANR EQU 1000 ;LIGHT FOR RIGHT UNIT	
845	0080	.	ANL EQU 2000 ;LIGHT FOR LEFT UNIT	
846	0000	.	*****	
847	0000	.	; INPDEV, OUTDEV, BXSTAT - I/O DEVICES *	
848	0000	.	*****	
849	0001	.	LFTCTU EQU 10 ;LEFT CARTRIDGE TAPE UNIT	
850	0002	.	RGTCTU EQU 20 ;RIGHT CARTRIDGE TAPE UNIT	
851	0004	.	DISPLY EQU 40 ;DISPLAY	
852	0008	.	PRINTR EQU 100 ;PRINTER	
853	0010	.	ALTIO EQU 200 ;ALTERNATE I/O	

13255

2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 24
=====
854      0020      . . .      DATCOM EQU   400      ;DATA COMM
855      0080      . . .      BUFBSY EQU  2000     ;BUF HELD BY UNSPECIFIED DEV
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 25
=====
857      FF54      . . .      SCNCNT EQU  CMND-1      ;NUM. OF KBSCAN PER CTU SCAN
858      FF53      . . .      CTBLNK EQU  SCNCNT-1    ;BLINK MASK FOR EJECT LIGHTS
859      FF52      . . .      CTBLTM EQU  CTBLNK-1    ;BLINK TIMER
860      0020      . . .      CTBDLY EQU  400         ;BLINK DELAY
861      FF51      . . .      HOLCNT EQU  CTBLTM-1    ;HOLE COUNTER
862      FF50      . . .      TPSTAL EQU  HOLCNT-1    ;TAPE STALL COUNTER
863      0000      . . .      ;*****
864      0000      . . .      ; I/O VARIABLES *
865      0000      . . .      ;*****
866      FF4F      . . .      IOCERR EQU  TPSTAL-1    ;I/O ERROR FLAG
867      0000      . . .      ;
868      0000      . . .      ;
869      FF4E      . . .      INPDEV EQU  IOCERR-1    ;CURRENT INPUT DEVICE
870      FF4D      . . .      OUTDEV EQU  INPDEV-1    ;CURRENT OUTPUT DEVICE
871      FF4C      . . .      IOCDPT EQU  OUTDEV-1    ;DEVICE FLAG POINTER
872      FF4B      . . .      IOSTA3 EQU  IOCDPT-1    ;DEVICE STATUS BYTE 3
873      FF4A      . . .      IOSTA2 EQU  IOSTA3-1    ;DEVICE STATUS BYTE 2
874      FF49      . . .      IOSTA1 EQU  IOSTA2-1    ;DEVICE STATUS BYTE 1
875      FF48      . . .      IOSTA0 EQU  IOSTA1-1    ;DEVICE NUMBER FOR STATUS
876      FF47      . . .      XFRLIM EQU  IOSTA0-1    ;TRANSFER LIMIT
877      FF46      . . .      CMLPIM EQU  XFRLIM-1    ;COMPARE LIMIT
878      FF3D      . . .      B2DBUF EQU  CMLPIM-9    ;BIN TO DECIMAL CONV BUFFER
879      003D      . . .      B2DBFL EQU  B2DBUF-BASE ;LSB PART OF "B2DBUF"
880      FF3C      . . .      B2DPTR EQU  B2DBUF-1    ;B2DBUF "GET" POINTER (LSB)
881      FF3B      . . .      B2DEND EQU  B2DPTR-1    ;B2DBUF END POINTER
882      0000      . . .      ;
883      0000      . . .      ; I/O CONTROL VARIABLES
884      0000      . . .      ;
885      FFD8      . . .      IOCDEV EQU  PARM1       ;DEVICE FLAG
886      FFDA      . . .      IOCOU EQU  PARM2       ;OUTPUT DEVICE ACCUMULATOR
887      FFD9      . . .      IOCINP EQU  PARM3       ;INPUT DEVICE ACCUMULATOR
888      FFD8      . . .      IOCTYP EQU  PARM4       ;COMMAND MODIFIER FLAG
889      FFD7      . . .      IOCMND EQU  PARM5       ;COMMAND TYPE FLAG
890      FFD5      . . .      IOCCNT EQU  PARM6       ;DATA COUNT (2 BYTES)
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  26
=====
892      0000      . . .      ;
893      0000      . . .      ; I/O BUFFER INFORMATION STORAGE
894      0000      . . .      ;
895      FF3A      . . .      B1STAT EQU  B2DEND-1 ;STATUS OF FIRST BUFFER
896      FF39      . . .      B1TYPE EQU  B1STAT-1 ;TYPE (-1=NORM, 0=EOF, 1=EVD
897      FF38      . . .      B1LEN  EQU  B1TYPE-1 ;LENGTH OF RECORD
898      FF37      . . .      B2STAT EQU  B1LEN-1  ;STATUS OF SECOND BUFFER
899      FF36      . . .      B2TYPE EQU  B2STAT-1 ;TYPE (-1=NORM, 0=EOF, 1=EVD
900      FF35      . . .      B2LEN  EQU  B2TYPE-1 ;LENGTH OF RECORD
901      0000      . . .      ;
902      0000      . . .      ; STORAGE FOR CARTRIDGE TAPE INTERRUPT ROUTINES
903      0000      . . .      ;
904      FF33      . . .      CTIADR EQU  B2LEN-2  ;ADDRESS (HAS SEVERAL USES)
905      FF31      . . .      CTISPT EQU  CTIADR-2 ;POINTER TO BUFFER STATUS
906      FF2F      . . .      CTIBPT EQU  CTISPT-2 ;POINTER TO BUFFER
907      FF2C      . . .      CTICNT EQU  CTIBPT-3 ;GENERAL COUNTERS
908      FF2B      . . .      CTITRL EQU  CTICNT-1 ;RE-READ COUNTER, HOLE CNTR
909      FF2A      . . .      CTICSM EQU  CTITRL-1 ;CHECKSUM COUNTER
910      FF29      . . .      CTISTA EQU  CTICSM-1 ;COMMAND SOURCE FLAG
911      0000      . . .      ;
912      0000      . . .      ; STORAGE FOR READ AND RECORD
913      0000      . . .      ;
914      FF27      . . .      NXTRED EQU  CTISTA-2 ;PTR INTO BUF FOR NEXT READ
915      FF25      . . .      LSTRED EQU  NXTRED-2 ;PTR INTO BUF FOR READ REPEA
916      FF24      . . .      SWPCTU EQU  LSTRED-1 ;SWAP CTU IN LOGGING MODE
917      0000      . . .      ;
918      0000      . . .      ;
919      FF23      . . .      SAVINP EQU  SWPCTU-1 ;"INPDEV" SAVE FOR LOCAL RCR
920      FF22      . . .      SAVOUT EQU  SAVINP-1 ;SAVE OUTDEV DURING LCL READ
921      0000      . . .      ;
922      0000      . . .      ; DATA FOR FORMAT DISPLAY STORAGE
923      0000      . . .      ;
924      FF21      . . .      ENDCOL EQU  SAVOUT-1 ;ENDING COLUMN AND ROW FOR
925      FF20      . . .      ENDROW EQU  ENDCOL-1 ;PREV NON-PROTECTED FIELD
=====

```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 27
=====
  927     0000     . . .      ;
  928     0000     . . .      ; EXTENDED MAIN CODE RAM AREA
  929     0000     . . .      ;
  930     FE80     . . .      XTRASP EQU 1772000
  931     0000     . . .      ;*****
  932     FE7F     . . .      DEVFLG EQU XTRASP-1 ;DEVICE PRESENT FLAG
  933     0000     . . .      ;*****
  934     0080     . . .      CTUIN EQU 2000      ;CTU CODE PRESENT
  935     0040     . . .      ALTIN EQU 1000     ;ALTERNATE I/O PRESENT
  936     0000     . . .      ;*****
  937     0000     . . .      ; PRINTER VARIABLES *
  938     0000     . . .      ;*****
  939     FE7D     . . .      PTRBBG EQU DEVFLG-2 ;START OF PRINTER BUFFER
  940     FE7B     . . .      PTRSPT EQU PTRBBG-2 ;LOAD POINTER
  941     FE79     . . .      PTRBPT EQU PTRSPT-2 ;UNLOAD POINTER
  942     FE78     . . .      PTRABT EQU PTRBPT-1 ;PRINTER ERROR FLAG
  943     0000     . . .      ;
  944     0000     . . .      ;
  945     FE77     . . .      PTRFLG EQU PTRABT-1 ;PRINTER TYPE FLAG
  946     0000     . . .      ;
  947     0000     . . .      ;
  948     0000     . . .      ;
  949     0000     . . .      ;*****
  950     0000     . . .      ; FLAGS FOR EXECUTING SOFT KEYS
  951     0000     . . .      ;*****
  952     FE76     . . .      SKFLGS EQU PTRFLG-1 ;SOFT KEY FLAGS
  953     0000     . . .      ;
  954     0001     . . .      DEFKEY EQU 10      ;SOFT KEY BEING DEFINED
  955     0002     . . .      SKIP EQU 20       ;SOFT KEY BEING EXECUTED
  956     0004     . . .      NEWKEY EQU 40     ;NEW SOFT KEY TRIGGERED
  957     0000     . . .      ;
  958     FE75     . . .      SFTKEY EQU SKFLGS-1 ;SOFT KEY BEING EXECUTED
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  28
=====
 960      0000      . . .      ;*****
 961      0000      . . .      ; ENTRY VECTORS TO I/O ROUTINES *
 962      0000      . . .      ;*****
 963      0000      . . .      ;
 964      0000      . . .      ;  KEYBOARD INITIATED FUNCTIONS
 965      0000      . . .      ;
 966      2800      . . .      IOORG EQU 240000 ;START OF I/O CODE
 967      2802      . . .      IOCKEY EQU IOORG+2 ;I/O CONTROL KEY
 968      2805      . . .      REDKEY EQU IOCKEY+3 ;READ KEY
 969      2808      . . .      CTLRED EQU REDKEY+3 ;CONTROL READ KEY
 970      2808      . . .      RECKEY EQU CTLRED+3 ;RECORD KEY
 971      280E      . . .      SELKEY EQU RECKEY+3 ;SELECT KEY
 972      2811      . . .      TSTCTU EQU SELKEY+3 ;CTU SELF-TEST
 973      2814      . . .      CONDTN EQU TSTCTU+3 ;CONDITION CARTRIDGE TAPES
 974      2817      . . .      RSTCTU EQU CONDTN+3 ;SOFT RESET FOR CTU
 975      0000      . . .      ;
 976      0000      . . .      ;  EXTERNALLY INITIATED FUNCTIONS
 977      0000      . . .      ;
 978      281A      . . .      IOCNTL EQU RSTCTU+3 ;I/O CONTROL ESCAPE SEQUENCE
 979      281D      . . .      IOSTGU EQU IOCNTL+3 ;SEND DEVICE STATUS
 980      2820      . . .      IODNGU EQU IOSTGU+3 ;SEND COMPLETION CODE
 981      2823      . . .      IORDGU EQU IODNGU+3 ;SEND I/O RECORD
 982      2826      . . .      RCRDGO EQU IORDGU+3 ;START REMOTE RECORD FUNCTIO
 983      2829      . . .      BNRYGO EQU RCRDGO+3 ;SEND BINARY DATA
 984      282C      . . .      CTDCDP EQU BNRYGO+3 ;SEND BINARY FILE
 985      0000      . . .      ;*****
 986      0000      . . .      ; INTERNAL ROUTINES *
 987      0000      . . .      ;*****
 988      282F      . . .      CTMON EQU CTDCDP+3 ;MONITOR CARTRIDGE DRIVES
 989      2832      . . .      PTTPLN EQU CTMON+3 ;PUT TOP LINE ONTO I/O DEV'S
 990      2835      . . .      DOOCTI EQU PTTPLN+3 ;INITIAL CTU INTERRUPT VECTO
 991      2837      . . .      RDABRT EQU DOOCTI+2 ;ABORT USER INITIATED READ
 992      283A      . . .      BSYCHK EQU RDABRT+3 ;WAIT UNTIL TAPE I/O DONE
 993      283D      . . .      CTINTR EQU BSYCHK+3 ;CTU INTERRUPT ROUTINE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
995	0000	.	.	;*****	29
996	0000	.	.	; TERMINAL START-UP *	
997	0000	.	.	;*****	
998	0000	.	.	ORG 00	
999	0000	.	.	BEGIN EQU \$	
1000	0000	54	.	DB VERSN ;ROM PRESENT FLAGS	
1001	0001	00	.	DB BEGIN/256 ;(= MOV D,B; NOP)	
1002	0002	F3	.	DI ;DISABLE INTERRUPTS	
1003	0003	3E	03	MVI A,SETROM+TMIEN+TMRON	
1004	0005	C3	F3 00	JMP GO ;GO TO START UP ROUTINE	
1005	0008	.	.	;*****	
1006	0008	.	.	; FIRMWARE INVOKED INTERRUPT *	
1007	0008	.	.	;*****	
1008	0008	E9	.	PCHL ;USE AS PCHL SUBROUTINE CALL	
1009	0009	.	.	ORG BEGIN+200	
1010	0010	.	.	;*****	
1011	0010	.	.	; TOP PLANE INTERRUPT 20B *	
1012	0010	.	.	;*****	
1013	0010	F5	.	PUSH PSW ;SAVE A-REGISTER AND FLAGS	
1014	0011	B7	.	ORA A ;CLEAR C-FLAG	
1015	0012	3E	32	MVI A,TWO ;SET INTERRUPT CODE	
1016	0014	C3	3D 17	JMP INTRPT ;HANDLE UNKNOWN INTERRUPTS	
1017	0017	.	.	ORG BEGIN+300	
1018	0018	.	.	;*****	
1019	0018	.	.	; TIMER INTERRUPT *	
1020	0018	.	.	;*****	
1021	0018	F5	.	PUSH PSW ;SAVE A-REGISTER, FLAGS	
1022	0019	C5	.	PUSH B ;AND REGISTER B AND C	
1023	001A	3E	33	MVI A,THREE ;SET INTERRUPT CODE	
1024	001C	C3	A9 08	JMP TMINTR ;CONTINUE TIMER ROUTINE	
1025	001F	.	.	ORG BEGIN+400	
1026	0020	.	.	;*****	
1027	0020	.	.	; DATA COMM INTERRUPT *	
1028	0020	.	.	;*****	
1029	0020	F5	.	PUSH PSW ;SAVE A-REGISTER AND FLAGS	
1030	0021	3E	34	MVI A,FOUR ;SET INTERRUPT CODE	
1031	0023	C3	65 13	JMP DCMINT ;CONTINUE INTERRUPT PROCESS	
1032	0026	.	.	ORG BEGIN+500	
1033	0028	.	.	;*****	
1034	0028	.	.	; I/O DEVICE INTERRUPT *	
1035	0028	.	.	;*****	
1036	0028	F5	.	PUSH PSW ;SAVE A-REG, STATUS	
1037	0029	E5	.	PUSH H ;AND H,L	
1038	002A	3E	35	MVI A,FIVE ;SET INTERRUPT CODE	
1039	002C	C3	FF 16	JMP IOINTR ;CONTINUE I/O ROUTINE	
1040	002F	.	.	ORG BEGIN+600	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 30
=====
1042     0030      . . .      ;*****
1043     0030      . . .      ; TOP PLANE INTERRUPT 60B *
1044     0030      . . .      ;*****
1045     0030     F5 . .      PUSH PSW          ;SAVE A-REGISTER AND FLAGS
1046     0031     B7 . .      URA A            ;CLEAR THE C-FLAG
1047     0032     3E 36 .    MVI A,SIX       ;SET INTERRUPT CODE
1048     0034     C3 3D 17   JMP INTRPT      ;HANDLE UNKNOWN INTERRUPTS
1049     0037      . . .      ORG BEGIN+700
1050     0038      . . .      ;*****
1051     0038      . . .      ; TEST POINT INTERRUPT *
1052     0038      . . .      ;*****
1053     0038     F5 . .      PUSH PSW          ;SAVE A-REGISTER AND FLAGS
1054     0039     B7 . .      URA A            ;CLEAR THE C-FLAG
1055     003A     3E 37 .    MVI A,SEVEN    ;SET INTERRUPT CODE
1056     003C     C3 3D 17   JMP INTRPT      ;HANDLE UNKNOWN INTERRUPTS
1057     003F      . . .      ORG BEGIN+1000
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1059	0040	.	.	*****	31
1060	0040	.	.	; VECTORS TO MAIN CODE ROUTINES *	
1061	0040	.	.	*****	
1062	0040	C3	33 1E	ZDSPMS JMP DSPMSG ;DISPLAY MESSAGE	
1063	0043	C3	6A 1E	JMP RSTDSP ;RESTORE NORMAL DISPLAY	
1064	0046	C3	93 13	JMP DCNUM ;ACCUMULATE DIGIT AND SIGN	
1065	0049	C3	87 13	JMP DCPLUS ;FOR PARAMETERIZED ESCAPE	
1066	004C	C3	8C 13	JMP DCMNUS ;SEQUENCES	
1067	004F	C3	48 05	JMP ESCEND ;TERMINATE ESCAPE SEQUENCE	
1068	0052	C3	42 11	JMP CHKLIM	
1069	0055	C3	9B 11	JMP CLBLXF	
1070	0058	C3	25 18	JMP SBLXF0	
1071	005B	C3	28 18	JMP SBLXFA ;KEYBOARD INITIATED BLK XFR	
1072	005E	C3	BE 18	JMP STRTBL ;START BLOCK RECORD	
1073	0061	C3	9D 1E	JMP CURPH ;HOME CURSOR (-XMIT ONLY)	
1074	0064	C3	38 12	JMP CURPHD ;CURSOR HOME DOWN	
1075	0067	C3	5C 16	JMP FRECNT ;CHECK NUMBER OF FREE BLOCKS	
1076	006A	C3	10 07	JMP PTBLK ;RELEASE BLOCKS FROM DISPLAY	
1077	006D	C3	95 1D	JMP CLEARL ;CLEAR LINE	
1078	0070	C3	C0 11	JMP CLEARS ;CLEAR DISPLAY FROM CURSOR	
1079	0073	C3	4C 16	JMP FNDB2 ;SET BIT N (B-REG = N)	
1080	0076	C3	4E 13	JMP SDTERM ;SEND TERMINATORS	
1081	0079	C3	51 18	JMP SDTRM1 ;SEND TERMINATOR ONLY	
1082	007C	C3	22 19	JMP XPUTDC ;TRANSMIT CHARACTER IN A-REG	
1083	007F	C3	9D 0E	JMP TRMTST ;TERMINAL SELF-TEST	
1084	0082	C3	C2 03	JMP CHINT0 ;EXECUTE CHARACTER FUNCTION	
1085	0085	C3	1F 7D	JMP INITD0 ;INIT FOR DISPLAY GET	
1086	0088	C3	A3 26	JMP GETDSP ;GET DISPLAY BYTE	
1087	008B	C3	6F 0B	JMP LNFEED ;DO LINE FEED	
1088	008E	C3	CA 25	JMP EXPAND ;EXPAND DISPLAY CONTROL CHAR	
1089	0091	C3	90 0C	JMP NXTCHR ;GET NEXT DISPLAY CHARACTER	
1090	0094	C3	88 05	JMP GETDCM ;PROCESS DATA COMM INPUT	
1091	0097	C3	ED 0B	JMP MLKSCO ;LOCATE FIRST UNLOCKED ROW	
1092	009A	C3	C2 0B	JMP MLKDF0 ;TURN OFF MEMORY LOCK	
1093	009D	C3	85 13	JMP HANGU0 ;HANG TERMINAL ON FATAL ERRO	
1094	00A0	S1	10 .	DW BUFMSG ;BUFFER OVERFLOW MESSAGE	
1095	00A2	C3	CA 13	JMP DCTEST ;DATA COMM SELF-TEST	
1096	00A5	C3	E5 16	JMP IORMG0 ;EXECUTE CODE FROM OPTION RO	
1097	00A8	C3	2E 09	JMP BN2DEC ;CONVERT BINARY TO DECIMAL	
1098	00AB	C3	1D 09	JMP BN2DE0 ;CONVERT SINGLE BYTE TO DEC	
1099	00AE	C3	A1 07	JMP RCADRA ;LOCATE CURSOR LOCATION	
1100	00B1	C3	84 11	JMP GTMODE ;CHECK FOR PAGE MODE	
1101	00B4	C3	E4 13	JMP DELAY ;DELAY 10MS * L REG	
1102	00B7	C3	11 05	JMP ESCAPE ;SET UP FOR ESCAPE SEQ	
1103	00BA	C3	41 24	JMP DISPC0 ;ADD CHARACTER ENHANCEMENT	
1104	00BD	C3	AF 21	JMP CRADV1 ;CLEAR CURSOR ADVANCE FLAG	
1105	00C0	C3	66 23	JMP CRRET ;DO A CARRIAGE RETURN	
1106	00C3	C3	D7 13	JMP DCXB2D ;SEE IF CHAR FROM KB (ZDC10)	
1107	00C6	C3	E5 1A	JMP CHKSEK ;SEE IF IN SOFT KEY MODE	
1108	00C9	C3	83 25	JMP DSPCHR ;ADD CHAR TO DISPLAY	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 32
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1109	00CC	C3	23 09	JMP BN2DA ;CONVERT & SEND A REG	
1110	00CF	C3	34 09	JMP B2DDE ;CONVERT & SEND DE REG	
1111	00D2	C3	95 11	JMP CKRMTE ;CHECK IF IN REMOTE MODE	
1112	00D5	C3	25 18	JMP SBLXF0 ;SET BLOCK XFER FLAG	
1113	00D8	C3	98 11	JMP CLBLXF ;CLEAR BLOCK XFER FLAG	
1114	00DB	C3	4E 13	JMP SDTERM ;SEND TERMINATOR	
1115	00DE	C3	E0 17	JMP PRMSEQ ;ESC & RECEIVED	
1116	00E1	C3	0A 24	JMP LOCLN2 ;PROCESS CHAR THRU LOCLIN	
1117	00E4	C3	2E 7D	JMP INITDG ;INIT. FOR DISPLAY GET	
1118	00E7	C3	22 18	JMP ENTREN ;ESC D--ENTER	
1119	00EA	C3	88 13	JMP HANGUP ;HANG TERMINAL	
1120	00ED	C3	39 1E	JMP DSPMG2 ;DISPLAY MESSAGE WITH G ON	
1121	00F0	C3	8B 04	JMP CHKCTL ;CHECK BLOCK XFER TRIGGER	

```
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 33
1123	00F3	.	.	*****	
1124	00F3	.	.	; TERMINAL RESET - START UP TERMINAL *	
1125	00F3	.	.	*****	
1126	00F3	.	.	GO EQU \$	
1127	00F3	D3	70	OUT PROCSR ;SET INITIAL PROCESSOR STATE	
1128	00F5	32	F5	STA PRCTL ;SET PROCESSOR STATE	
1129	00F8	31	60	LXI SP,STACK ;SET STACK POINTER	
1130	00FB	3A	CD	LDA ECONTF	
1131	00FE	FE	C3	CPI JMP ;FIRST TURN ON?	
1132	0100	.	.	*****	
1133	0100	C2	39	JNZ INIT ;YES - INITIALIZE TERMINAL	
1134	0103	.	.	*****	
1135	0103	3A	F8	LDA CMFLGS ;NO - GET COMMON FLAGS	
1136	0106	E6	04	ANI FRCRST ;FORCE FULL RESET?	
1137	0108	C2	39	JNZ INIT ;YES - INITIALIZE TERMINAL	
1138	0108	21	D0	LXI H,RSTTMR ;NO - GET SOFT RESET TIMER	
1139	010E	B6	.	ORA M ;FULL RESET ACTIVE?	
1140	010F	CA	17	JZ G0010 ;NO - START SOFT RESET	
1141	0112	FE	32	CPI SFTDLY ;STILL IN SOFT RESET START?	
1142	0114	.	.	; (CAUSED BY CONTACT BOUNCE)	
1143	0114	C2	39	JNZ INIT ;NO - DO FULL RESET	
1144	0117	.	.	G0010 EQU \$;YES - RESTART SOFT RESET	
1145	0117	36	32	MVI M,SFTDLY ;NO - SET 0.5 SEC TIME OUT	
1146	0119	.	.	*****	
1147	0119	.	.	; DO SOFT RESET *	
1148	0119	.	.	*****	
1149	0119	.	.	G01 EQU \$;ENTRY FOR SOFT RESET	
1150	0119	32	6E	STA DFLGS ;CLEAR DATA TRANSFER FLAGS	
1151	011C	32	76	STA SKFLGS ;CLEAR SOFT KEY FLAGS	
1152	011F	3E	07	MVI A,RSETKB	
1153	0121	CD	08	CALL ZK9CTL ;RESET THE KEYBOARD	
1154	0124	3E	02	MVI A,RSETDC	
1155	0126	CD	7E	CALL DCMCT1 ;RESET THE DATA COMM	
1156	0129	.	.	;!!!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!!!*	
1157	0129	CD	05	CALL ZGSOFT ;RESET GRAPHICS	
1158	012C	.	.	*****	
1159	012C	21	17	LXI H,RSTCTU ;RESET CARTRIDGE TAPES	
1160	012F	CD	E5	CALL IORMGO ;IF CTU CODE PRESENT	
1161	0132	CD	6A	CALL RSTDSP ;RESTORE NORMAL DISPLAY	
1162	0135	FB	.	EI ;ENABLE INTERRUPTS	
1163	0136	C3	60	JMP START ;NO, USE NORMAL RANGE TABLE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 34
=====
1165     0139      . . .      ;*****
1166     0139      . . .      ; INIT - DO COMPLETE TERMINAL INITIALIZATION *
1167     0139      . . .      ;*****
1168     0139      . . .      INIT EQU $
1169     0139      AF . . .      XRA A          ;CLEAR TO ZERO
1170     013A      32 CD FF      STA ECONTF     ;CLEAR "JMP" TO FORCE FULL
1171     0130      . . .      ;
1172     0130      . . .      ;*****
1173     0130      . . .      ; CLEAR FAST RAM ON BOTH ROM BOARDS
1174     0130      11 00 90      LXI D,FSTRAM-256 ;GRAPHICS FAST RAM
1175     0140      21 00 91      LXI H,FSTRAM   ;2645 FAST RAM
1176     0143      . . .      INI010 EQU $
1177     0143      77 . . .      MOV M,A        ;CLEAR A BYTE
1178     0144      12 . . .      STAX D         ;ON BOTH BOARDS
1179     0145      1C . . .      INR E         ;UPDATE ADDRESSES
1180     0146      2C . . .      INR L
1181     0147      C2 43 01      JNZ INI010     ;LOOP TILL ALL 256 DONE
1182     014A      . . .      ;*****
1183     014A      . . .      ;
1184     014A      . . .      ; CLEAR SLOW RAM AREA
1185     014A      . . .      ;
1186     014A      5D . . .      MOV E,L        ;SET E = 0 FOR 256 BYTES
1187     014B      . . .      ;*****
1188     014B      . . .      ; CLEAR ALL SLOW RAM VARIABLES
1189     014B      26 F7 .      MVI H,DSPLIM/256 ;START ADDRESS
1190     014D      . . .      ;*****
1191     014D      . . .      INI020 EQU $
1192     014D      CD 30 12      CALL CLRAL1    ;CLEAR A 256 BYTE SECTION
1193     0150      BC . . .      CMP H         ;ALL SECTIONS CLEARED?
1194     0151      C2 4D 01      JNZ INI020     ;NO - CONTINUE CLEARING
1195     0154      . . .      ;*****
1196     0154      . . .      ; LOCATE NON-DISPLAY SPACE *
1197     0154      . . .      ;*****
1198     0154      21 FF CF      LXI H,BFSPCE  ;SET UPPER BOUNDARY ADDRESS
1199     0157      22 8B FF      SHLD BUFEND    ;OF NON-DISPLAY BUFFER ARE
1200     015A      06 B0 .      MVI B,LWBUF    ;SET B TO MSB OF LOWER LIMIT
1201     015C      CD 6C 05      CALL FNDRAM
1202     015F      22 8D FF      SHLD BUFBN     ;STORE BUFFER START ADDRESS
1203     0162      . . .      ;
1204     0162      . . .      ; LOCATE DISPLAY SPACE
1205     0162      . . .      ;
1206     0162      21 FF F7      LXI H,DSPLIM  ;SET UPPER BOUNDARY ADDRESS
1207     0165      22 A8 FF      SHLD DSPEND    ;OF DISPLAY AREA
1208     0168      06 D0 .      MVI B,LWDSP    ;SET B TO MSB OF LOWER LIMIT
1209     016A      CD 6C 05      CALL FNDRAM
1210     016D      22 AA FF      SHLD DSPBN     ;STORE DISPLAY START ADDRESS
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 35
1212	0170	.	.	*****	
1213	0170	.	.	; INITIALIZE PROCESSOR BOARD STATE, KEYBOARD, *	
1214	0170	.	.	; AND DATA COMM *	
1215	0170	.	.	*****	
1216	0170	3E	03	MVI A,SETROM+TMIEN+TMRON	
1217	0172	32	F5	STA PRCTL ;ENABLE ROM'S AND TIMER	
1218	0175	3E	C9	MVI A,RET ;PUT RETURN CODE INTO	
1219	0177	32	65	STA INTVEC ;INTERRUPT VECTOR AND	
1220	017A	32	68	STA SCNVEC ;DISPLAY SCAN VECTOR	
1221	017D	.	.	;	
1222	017D	.	.	;	
1223	017D	.	.	;	
1224	017D	.	.	;	
1225	017D	.	.	;	
1226	017D	CD	02	CALL ZINIKB ;SET JUMPERS AND DC SWITCHES	
1227	0180	CD	08	CALL ZINIDC ;FETCH BUFFER REQUIREMENTS	
1228	0183	CD	87	CALL GETBUF ;ALLOCATE BUFFER SPACE	
1229	0186	CD	0B	CALL ZIN2DC ;COMPLETE DATA COMM INIT	
1230	0189	DA	85	JC HANGUO ;(PROCESS ERROR IF ANY)	
1231	018C	.	.	;	
1232	018C	.	.	;	
1233	018C	21	FD	LXI H,TRMTYP	
1234	018F	36	04	MVI M,HP2648	
1235	0191	.	.	;	
1236	0191	.	.	;	
1237	0191	.	.	;	
1238	0191	.	.	;	
1239	0191	21	02	LXI H,1*256+2 ;OUTPUT = RIGHT CTU (2)	
1240	0194	22	4D	SHLD OUTDEV ;INPUT = LEFT CTU (1)	
1241	0197	2A	35	LHLD DOOCTI ;SET INITIAL CARTRIDGE TAPE	
1242	019A	22	E1	SHLD CTIVEC ;INTERUPT VECTOR	
1243	019D	3E	C3	MVI A,JMP ;SET JUMP COMMAND FOR	
1244	019F	32	E0	STA CTIJMP ;CTU INTERRUPT VECTOR	
1245	01A2	.	.	;	
1246	01A2	.	.	;	
1247	01A2	.	.	;	
1248	01A2	21	02	LXI H,ZINIAL ;INITIALIZE ALTERNATE I/O	
1249	01A5	CD	E5	CALL IORMGO ;DEVICE	
1250	01A8	3E	00	MVI A,0 ;(SET FOR NO ALTERNATE I/O	
1251	01AA	DA	B5	JC INI110 ;BYPASS INIT IF NO ALT I/O	
1252	01AD	CD	87	CALL GETBUF ;ELSE, ALLOCATED BUFFER	
1253	01B0	CD	05	CALL ZIN2AL ;AND CONTINUE INIT	
1254	01B3	3E	40	MVI A,ALTIN ;SET ALT I/O PRESENT BIT	
1255	01B5	.	.	INI110 EQU \$	
1256	01B5	47	.	MOV B,A ;SAVE ALTERNATE I/O STATUS	
1257	01B6	21	00	LXI H,IOORG ;SET I/O START ADDRESS	
1258	01B9	CD	F5	CALL IORMG1 ;DOES I/O CODE EXIST?	
1259	01BC	78	.	MOV A,B ;(GET CURRENT I/O OPTIONS)	
1260	01BD	C2	C6	JNZ INI130 ;NO - DON'T SET I/O BIT	
1261	01C0	F6	80	ORI CTUIN ;ELSE SET CTU PRESENT BIT	

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 36
=====
1262     01C2     21  FD  FF          LXI  H,TRMTYP ;SET TERM TYPE TO INDICATE
1263     01C5     34  .   .          INR  M          ;I/O CODE INCLUDED
1264     01C6     .   .   .          INI130 EQU  $
1265     01C6     32  7F  FE          STA  DEVFLG    ;SET I/O OPTIONS FLAG
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 37
1267	01C9	.	.	*****	
1268	01C9	.	.	; BUFFER ALLOCATIONS FOR DATACOM, ALT. I/O MAY	
1269	01C9	.	.	; HAVE USED UP ALL OF THE DISPLAY MEMORY	
1270	01C9	.	.	; SEE IF THERE IS A MINIMUM AMOUNT LEFT	
1271	01C9	2A	A8	FF LHL D DSPEND ;SAVE DSPEND	
1272	01CC	E5	.	. PUSH H	
1273	01CD	01	00	02 LXI B,MINMEM ;SEE IF MIN. AMOUNT IS LEFT	
1274	01D0	CD	99	05 CALL GTR010 ;IN DISPLAY MEMORY	
1275	01D3	E1	.	. POP H ;YES--RESTORE DSPEND	
1276	01D4	22	A8	FF SHLD DSPEND	
1277	01D7	.	.	*****	
1278	01D7	.	.	*****	
1279	01D7	.	.	; GENERATE FREE BLOCKS LIST FOR DISPLAY *	
1280	01D7	.	.	*****	
1281	01D7	11	F1	FF LXI D,1-BLKSZ ;ADDRESSED DISPLAY BLOCK	
1282	01DA	19	.	. DAD D	
1283	01DB	7D	.	. MOV A,L ;COMPUTE ADDRESS OF LSB PART	
1284	01DC	F6	0F	. ORI BLKSM ;OF PREVIOUS LINE POINTER	
1285	01DE	6F	.	. MOV L,A ;IN HIGHEST ADDRESSED	
1286	01DF	2B	.	. DCX H ;DISPLAY BLOCK	
1287	01E0	36	00	. MVI M,0 ;SET IT TO ZERO TO INDICATE	
1288	01E2	2B	.	. DCX H ;END OF FREE LIST	
1289	01E3	EB	.	. XCHG ;SET NEXT BLOCK LINK OF	
1290	01E4	2A	AA	FF LHL D DSPBGN ;LOWEST ADDRESSED DISPLAY	
1291	01E7	73	.	. MOV M,E ;BLOCK TO POINT TO MSB	
1292	01E8	23	.	. INX H ;PART OF NEXT LINE LINK IN	
1293	01E9	72	.	. MOV M,D ;HIGHEST BLOCK	
1294	01EA	EB	.	. XCHG ;SWAP HIGH AND LOW ADDRESSES	
1295	01EB	13	.	. INX D ;ADJUST LOW ADDR TO LOW LIM	
1296	01EC	.	.	. ; FOR LINKING DISPLAY BLOCKS	
1297	01EC	2B	.	. DCX H ;SET FREE BLOCKS HEAD TO LSB	
1298	01ED	22	AC	FF SHLD FRBLKS ;PART OF NEXT LINE POINTER	
1299	01F0	.	.	. ; IN HIGHEST BLOCK	
1300	01F0	D6	0E	. SUI BLKSZ-2 ;SET B,L TO ADDRESS OF MSB	
1301	01F2	44	.	. MOV B,H ;PART OF NEXT BLOCK POINTE	
1302	01F3	6F	.	. MOV L,A ;IN HIGHEST DISPLAY BLOCK	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 38
=====
1304     01F4      . . .      ;*****
1305     01F4      . . .      ; CHAIN FREE BLOCKS *
1306     01F4      . . .      ;*****
1307     01F4      . . .      ;
1308     01F4      . . .      ; B,A = ADDRESS OF UPPER BYTE IN NEXT LOWER BLOCK
1309     01F4      . . .      ; D,E = LOWER LIMIT OF DISPLAY AREA
1310     01F4      . . .      ; H,L = ADDRESS OF MSB PART OF NEXT BLOCK LINK
1311     01F4      . . .      ;     IN CURRENT BLOCK
1312     01F4      . . .      ;
1313     01F4      . . .      INI210 EQU $
1314     01F4      7D . .      MOV  A,L      ;COMPUTE ADDRESS OF UPPERMOST
1315     01F5      D6 02 .      SUI  2      ;BYTE IN NEXT LOWER BLOCK
1316     01F7      D2 FB 01     JNC  INI220
1317     01FA      05 . .      DCR  B
1318     01FB      . . .      INI220 EQU $
1319     01FB      70 . .      MOV  M,B      ;LINK CURRENT BLOCK TO NEXT
1320     01FC      2B . .      DCX  H      ;LOWER BLOCK
1321     01FD      77 . .      MOV  M,A
1322     01FE      D6 0E .      SUI  BLKSZ-2 ;SET H,L TO ADDRESS OF MSB
1323     0200      6F . .      MOV  L,A      ;PART OF NEXT BLOCK LINK I
1324     0201      60 . .      MOV  H,B      ;NEXT LOWER BLOCK
1325     0202      93 . .      SUB  E      ;COMPARE AGAINST LOWER LIMIT
1326     0203      78 . .      MOV  A,B
1327     0204      9A . .      SBB  D      ;DISPLAY AREA EXHAUSTED?
1328     0205      D2 F4 01     JNC  INI210 ;NO - CONTINUE LINKING BLOCK
1329     0208      . . .      ; YES - SET UP INITIAL DISPLAY
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1331	0208	.	.	;	39
1332	0208	.	.	; SET UP INITIAL SOFT KEYS DISPLAY	
1333	0208	.	.	;	
1334	0208	CD	AB 06	CALL INITDS ;START A NEW DISPLAY LIST	
1335	0208	2B	.	DCX H ;SET SOFT KEY START ADDRESS	
1336	020C	22	A6 FF	SHLD SFTKYS ;TO FIRST CHARACTER	
1337	020F	3E	80 .	MVI A,CRTOFF ;SET CURRENT AND LAST ROW	
1338	0211	32	C0 FF	STA CURROW ;TO CONTROL FOR DISPLAY OF	
1339	0214	32	C7 FF	STA LSTROW	
1340	0217	.	.	*****	
1341	0217	.	.	; SET UP DEFINATION FOR SUFT RETURN KEY = F0	
1342	0217	21	2D 16	LXI H,RTNKEY ;POINTER TO DEFINITION STRIN	
1343	021A	CD	02 11	CALL XMS2DS ;TRANSFER TO DISPLAY	
1344	021D	.	.	*****	
1345	021D	.	.	;	
1346	021D	.	.	; SET UP KEY DEFINITIONS	
1347	021D	.	.	;	
1348	021D	01	4E FE	LXI B,DSPSTR-1	
1349	0220	21	1E 16	LXI H,ATBLIN ;TRANSFER ATTRIBUTE LINE	
1350	0223	CD	29 0C	CALL MOVCHR	
1351	0226	.	.	***** GRAPHICS MODIFICATION *****	
1352	0226	0E	08 .	MVI C,NMFCTK-1 ;# OF KEYS TO DEFINE	
1353	0228	.	.	*****	
1354	0228	.	.	;	
1355	0228	.	.	; BUILD ATTRIBUTE LINE	
1356	0228	.	.	;	
1357	0228	.	.	INI310 EQU \$	
1358	0228	.	.	***** GRAPHICS MODIFICATION *****	
1359	0228	3E	39 .	MVI A,ZERO+NMFCTK ;GET FUNCTION KEY NUMBE	
1360	022A	.	.	*****	
1361	022A	91	.	SUB C	
1362	0228	32	43 FE	STA DSPSTR-ATBLEN+2	
1363	022E	.	.	;	
1364	022E	.	.	; BUILD DEFINITION LINE	
1365	022E	.	.	;	
1366	022E	3E	78 .	MVI A,SMALLX ;COMPUTE CHAR AFTER <ESC>	
1367	0230	91	.	SUB C ;(LOWER CASE <P>-<W>)	
1368	0231	21	4D FE	LXI H,DSPSTR-CHRLOC	
1369	0234	77	.	MOV M,A ;SET DATA CHARACTER	
1370	0235	2E	41 .	MVI L,DSPSTR-ATBLEN-BASE2	
1371	0237	C5	.	PUSH B ;TRANSFER SOFT KEY DEFINITIU	
1372	0238	CD	02 11	CALL XMS2DS ;TU DISPLAY MEMORY	
1373	0238	C1	.	POP B	
1374	023C	0D	.	DCR C ;ALL KEYS DEFINED?	
1375	023D	C2	28 02	JNZ INI310 ;NO - DU NEXT KEY	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 40
=====
1377      0240      . . .      ;*****
1378      0240      . . .      ; SOFT KEYS DONE - SET INITIAL DISPLAY STATE *
1379      0240      . . .      ;*****
1380      0240      AF . .      XRA A      ;CLEAR LAST LINE POINTER
1381      0241      32 A1 FF      STA LLINE
1382      0244      3D . .      DCR A      ;SET DISPLAY TYPE TO SOFT
1383      0245      32 AE FF      STA DSPTYP ;KEY DISPLAY
1384      0248      CD 9D 1E      CALL CURPH ;HOME THE CURSOR
1385      0248      CD 17 23      CALL SWAP  ;SAVE SOFT KEY PARAMETERS
1386      024E      . . .      ;*****
1387      024E      . . .      ; INITIALIZE FIRST LINE OF DISPLAY *
1388      024E      . . .      ;*****
1389      024E      CD AB 06      CALL INITDS ;START A NEW DISPLAY LIST
1390      0251      . . .      ;***** GRAPHICS MODIFICATION *****
1391      0251      CD 02 60      CALL ZINGR  ;INITIALIZE GRAPHICS
1392      0254      . . .      ;*****
1393      0254      . . .      ;*****
1394      0254      . . .      ; PATCH TO FIX SOFT RESET BUG. RS232 PRINTER
1395      0254      . . .      ; CARD MUST BE RE-INITIALIZED AFTER
1396      0254      . . .      ; SOFT RESET
1397      0254      21 94 10      LXI H,TRMRDY ;PUT UP 'TERMINAL READY'
1398      0257      87 . .      URA A      ;ADD TO DISPLAY
1399      0258      CD 30 1E      CALL DSPMS1 ;MESSAGE
1400      025B      3E C3 .      MVI A,JMP  ;SET JUMP COMMAND FOR
1401      025D      32 CD FF      STA ECONTF ;CHINT
1402      0260      . . .      ;*****
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1404	0260	.	.	START EQU \$	41
1405	0260	.	.	;*****	
1406	0260	.	.	; INITIALIZE I/O DEVICES *	
1407	0260	.	.	;*****	
1408	0260	.	.	;	
1409	0260	.	.	; PRINTER INITIALIZATION ROUTINE	
1410	0260	.	.	;	
1411	0260	.	.	; CHECK FOR 9866 PRINTER FIRST	
1412	0260	.	.	;	
1413	0260	3A	00 8D	LDA PTRST1 ;GET STATUS FROM 9866 PCA	
1414	0263	B7	.	ORA A ;IS INTERFACE INSTALLED?	
1415	0264	CA	6F 02	JZ PTRI10 ;NO - LOOK FOR RS-232 PRNTR	
1416	0267	3A	02 8D	LDA PTRCL1 ;YES - CLEAR THE PRINTER	
1417	026A	3E	01 .	MVI A,1 ;SET PRINTER FLAG FOR	
1418	026C	C3	81 02	JMP PTR120 ;9866 PRINTER (= 1)	
1419	026F	.	.	;	
1420	026F	.	.	; RS-232 PRINTER 2	
1421	026F	.	.	;	
1422	026F	.	.	PTRI10 EQU \$	
1423	026F	3A	20 85	LDA PTRST2 ;GET STATUS FROM RS-232 PCA	
1424	0272	B7	.	ORA A ;IS RS-232 PCA INSTALLED?	
1425	0273	CA	81 02	JZ PTR120 ;NO - SET FOR NO PRINTER	
1426	0276	.	.	;	
1427	0276	3A	40 85	LDA PTRCF2 ;YES - GET CONFIG. STRAPS	
1428	0279	E6	1F .	ANI PTRBD2 ;ISOLATE BAUD AND PARITY	
1429	027B	17	.	RAL ;ADJUST FOR CONTROL OUTPUT	
1430	027C	32	40 85	STA PTROT2 ;SET BOARD TO CONFIGURATION	
1431	027F	3E	02 .	MVI A,2 ;SET FLAG FOR RS-232 PRINTER	
1432	0281	.	.	;	
1433	0281	.	.	PTR120 EQU \$	
1434	0281	32	77 FE	STA PTRFLG ;SET PRINTER FLAG	
1435	0284	.	.	;*****	
1436	0284	.	.	; INITIALIZE FLAGS AND RANGE TABLE ADDRESS *	
1437	0284	.	.	;*****	
1438	0284	CD	48 05	CALL ESCEND ;RESET NORMAL RANGE TABLE	
1439	0287	3A	29 48	LDA FRSALT ;SET INITIAL ALTERNATE	
1440	028A	32	72 FF	STA CHRSET ;CHARACTER SET	
1441	028D	CD	AF 21	CALL CRADV1 ;CLEAR CURSOR ADVANCE FLAG	
1442	0290	3D	.	DCR A ;CLEAR SPOW LATCH	
1443	0291	32	6C FF	STA SPOWL	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  42
=====
1445      0294      . . .      ;
1446      0294      . . .      ; WAIT LOOP
1447      0294      . . .      ;
1448      0294      . . .      WTL00P EQU $
1449      0294      31 60 91      LXI SP,STACK ;SET STACK POINTER
1450      0297      CD 10 06      CALL GETDC1 ;SET DISPLAY CURSOR
1451      029A      . . .      ;*****
1452      029A      . . .      ; CHECK FOR DATA COMM INPUT *
1453      029A      . . .      ;*****
1454      029A      . . .      WTL010 EQU $
1455      029A      CD B8 05      CALL GETDCM ;GET DATA COMM INPUT IF ANY
1456      029D      . . .      ;*****
1457      029D      . . .      ; CHECK FOR KEYBOARD INPUT *
1458      029D      . . .      ;*****
1459      029D      3A 6F FF      LDA MFLGS2 ;GET MODE FLAGS
1460      02A0      E6 08 .      ANI ESCINP ;ESCAPE SEQUENCE LOCK OUT?
1461      02A2      C2 B8 02      JNZ WTL020 ;YES - IGNORE KEYBOARD
1462      02A5      CD 05 48      CALL ZGETKY ;ANY KEYBOARD INPUT?
1463      02A8      CA D0 02      JZ WTL200 ;YES - PROCESS IT
1464      02AB      . . .      ;
1465      02AB      . . .      ; IF KEYBOARD LOCKED, A = CHARACTER HIT, IF ANY
1466      02AB      . . .      ; OTHERWISE A = 377B
1467      02AB      . . .      ;
1468      02AB      . . .      ;***** GRAPHICS MODIFICATION *****
1469      02AB      FE EF .      CPI SFTCR ;SOFT RETURN KEY?
1470      02AD      . . .      ;*****
1471      02AD      C2 B8 02      JNZ WTL020 ;NO - CHECK CTU & DISPATCHER
1472      02B0      3A 65 FF      LDA IOFLGS ;USER READ OR FILE READ
1473      02B3      E6 06 .      ANI USREAD+FILRED ;PENDING?
1474      02B5      C4 37 28      CNZ RDABRT ;YES - ABORT READ KEY
=====

```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 43
=====
1476     02B8      . . .      ;*****
1477     02B8      . . .      ; CHECK CTU'S AND PENDING BLOCK TRANSFERS *
1478     02B8      . . .      ;*****
1479     02B8      . . .      WTL020 EQU $
1480     02B8      . . .      ;!!!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!*
1481     02B8      CD 17 60    CALL ZVR          ;DO VERTICAL RETRACE SCAN
1482     02B8      . . .      ;*****
1483     02B8      21 54 FF    LXI H,SCNCNT    ;DECREMENT SCAN COUNT
1484     02BE      35 . .     DCR M           ;11 SCANS DONE?
1485     02BF      F2 9A 02    JP WTL010       ;NO - RESTART DO NOTHING LOO
1486     02C2      36 0A .     MVI M,10       ;YES - RESET SCAN COUNT
1487     02C4      CD 68 91    CALL SCNVEC     ;DO OPTIONAL DISPLAY SCAN
1488     02C7      CD 08 16    CALL IOCTMN    ;MONITOR TAPE DRIVES
1489     02CA      CD 02 04    CALL DSPTCH    ;CHECK PENDING BLOCK XFRS
1490     02CD      C3 9A 02    JMP WTL010     ;RESTART DO NOTHING LOOP
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  44
=====
1492      02D0      . . .      ;
1493      02D0      . . .      ; KEY HIT - CHECK FOR FUNCTION KEY
1494      02D0      . . .      ;
1495      02D0      . . .      WTL200 EQU $
1496      02D0      32 9C FF    STA CHARIN      ;SAVE KEYBOARD CHARACTER
1497      02D3      4F . .     MOV C,A         ;SAVE THE BYTE IN C-REGISTER
1498      02D4      3E FE .    MVI A,3779-SDACOM
1499      02D6      CD 53 17   CALL CLRDFL     ;CLEAR DATA COMM INPUT FLAG
1500      02D9      3A F8 FF   LDA CMFLGS     ;GET COMMON FLAGS
1501      02DC      2F . .     CMA            ;BOTH RECEIVE MODE FLAG SET
1502      02D0      E6 30 .    ANI RCVMODE+REMSET ;AND REMOTE ENABLED?
1503      02DF      C2 E8 02   JNZ WTL205     ;NO - PROCESS KEYBOARD INPUT
1504      02E2      CD 14 48   CALL ZBELL     ;YES - SOUND BELL AND
1505      02E5      C3 9A 02   JMP WTL010     ;IGNORE KEY
1506      02E8      . . .      ;
1507      02E8      . . .      WTL205 EQU $
1508      02E8      AF . .     XRA A          ;NO - PROCESS THE KEY
1509      02E9      57 . .     MOV D,A        ;(SET A,D = 0)
1510      02EA      B1 . .     ORA C          ;FUNCTION KEY?
1511      02EB      F2 7C 03   JP WTL300     ;NO - PROCESS ASCII KEY
1512      02EE      . . .      ;!!!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!!!
1513      02EE      . . .      ; TEST FOR GRAPHICS KEYPAD FUNCTION
1514      02EE      FE 98 .    CPI GFUNMX     ;GRAPHICS FUNCTION CODE?
1515      02F0      D2 F9 02   JNC WTL207     ;NO, CHECK FOR TABLE FUNC
1516      02F3      CD 2C 60   CALL ZGFUNC    ;YES, EXECUTE FUNCTION
1517      02F6      C3 94 02   JMP WTL00P    ;RESTART WAITLOOP
1518      02F9      . . .      WTL207 EQU $
1519      02F9      . . .      ;*****
1520      02F9      FE A1 .    CPI FNCLIM     ;TABLE FUNCTION?
1521      02FB      F2 0D 03   JP WTL210     ;NO - CHECK FOR F1-F8
1522      02FE      D6 98 .    SUI FNCLWR    ;COMPUTE TABLE INDEX
1523      0300      87 . .     ADD A
1524      0301      5F . .     MOV E,A        ;COMPUTE TABLE ADDRESS
1525      0302      21 00 16   LXI H,FNCTAB  ;(D = 0)
1526      0305      19 . .     DAD D
1527      0306      CD C6 1A   CALL CHAIN     ;GET THE FUNCTION ADDRESS
1528      0309      CF . .     RST RSTJMP    ;GO PERFORM FUNCTION
1529      030A      C3 94 02   JMP WTL00P    ;RESTART WAIT LOOP
1530      030D      . . .      ;
1531      030D      . . .      ; CHECK FOR F1-F8 KEY
1532      030D      . . .      ;
1533      030D      . . .      WTL210 EQU $
1534      030D      . . .      ;*****
1535      030D      FE EF .    CPI FOCODE     ;IS THE KEY F0-F8?
1536      030F      . . .      ;*****
1537      030F      DA 45 03   JC WTL250     ;NO - EXPAND ESCAPE SEQUENCE
1538      0312      . . .      ;***** GRAPHICS MODIFICATION *****
1539      0312      FE F8 .    CPI FOCODE+NMFCTK
1540      0314      . . .      ;*****
1541      0314      D2 45 03   JNC WTL250     ;NO - EXPAND ESCAPE SEQUENCE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
					45
1542	0317	.	.	*****	
1543	0317	.	.	; IF SOFT KEYS OR AUTO PLOT MENU UP, AND KEY IS	
1544	0317	.	.	; SOFT RETURN, REPLACE WITH HARD RETURN (15B)	
1545	0317	FE	EF	CPI SFTCR ;SOFT RETURN KEY?	
1546	0319	CA	25	03 JZ WTL220 ;YES, CHECK SOFT KEYS, A.P.	
1547	031C	CD	E5	1A CALL CHKSFK ;NO. ARE SOFT KEYS UP?	
1548	031F	.	.	; MUST EXECUTE SOFT KEY BY USING TRIGGER SOFT KEY	
1549	031F	.	.	; ROUTINE	
1550	031F	CA	31	03 JZ WTL225 ;IF NOT, EXECUTE KEY	
1551	0322	C3	94	02 JMP WTLOOP ;RETURN TO WAITLOOP	
1552	0325	.	.	WTL220 EQU \$	
1553	0325	CD	E5	1A CALL CHKSFK ;SOFT KEYS UP?	
1554	0328	C2	3C	03 JNZ WTL230 ;YES, DO A HARD RETURN	
1555	0328	CD	23	60 CALL ZMUCHK ;NO. AUTO PLOT MENU UP?	
1556	032E	C2	3C	03 JNZ WTL230 ;YES, DO A HARD RETURN	
1557	0331	.	.	; EXECUTE SOFT KEY (C = KEYCODE)	
1558	0331	.	.	WTL225 EQU \$	
1559	0331	3E	02	. MVI A,SKIP ;SET SOFT KEY IN PROGRESS	
1560	0333	CD	FD	23 CALL STSKFL ;FLAG	
1561	0336	CD	B8	23 CALL EXSK1 ;EXECUTE THE SOFT KEY	
1562	0339	C3	94	02 JMP WTLOOP ;RETURN TO WAITLOOP	
1563	033C	.	.	WTL230 EQU \$	
1564	033C	3E	0D	. MVI A,CR ;REPLACE SOFT WITH HARD	
1565	033E	4F	.	. MOV C,A ;RETURN	
1566	033F	32	9C	FF STA CHARIN ;USE CR AS CURRENT CHAR	
1567	0342	C3	7C	03 JMP WTL300 ;PROCESS HARD RETURN	
1568	0345	.	.	*****	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 46
=====
1570      0345      . . .      ;*****
1571      0345      . . .      ; PROCESS FUNCTION KEYS *
1572      0345      . . .      ;*****
1573      0345      . . .      WTL250 EQU $
1574      0345      0E 1B .      MVI C,ESC      ;SET ESCAPE AS INPUT CHAR
1575      0347      . . .      WTL260 EQU $
1576      0347      . . .      ;*****
1577      0347      . . .      ; NOTE THAT IF IN GRAPHICS GIN MODE, 2ND CHAR
1578      0347      . . .      ; OF LOCAL 2 CHAR ESCAPE SEQUENCE IS IGNORED
1579      0347      . . .      ;*****
1580      0347      CD C4 06      CALL LOCLIO     ;PROCESS KEYBOARD INPUT
1581      034A      21 9C FF      LXI H,CHARIN   ;RECALL KEYBOARD INPUT
1582      034D      7E . .      MOV A,M
1583      034E      FE FF .      CPI ENHNCF     ;DISPLAY ENHANCEMENT CODE?
1584      0350      CA 68 03      JZ WTL270      ;YES - EXPAND INTO AMPERSAND
1585      0353      FE FE .      CPI STF0R1     ;ENTER FOREIGN MODE CONTROL?
1586      0355      CA 6F 03      JZ WTL280      ;YES - CONTINUE SEQUENCE
1587      0358      FE FD .      CPI STF0R2     ;COMPLETE FOREIGN MODE SET?
1588      035A      CA 75 03      JZ WTL290      ;YES - SET ENDING SEQUENCE
1589      035D      E6 7F .      ANI 1770      ;NO - MASK OUT UPPER BIT
1590      035F      BE . .      CMP M          ;FUNCTION COMPLETED?
1591      0360      CA 94 02      JZ WTL00P      ;YES - RESTART WAIT LOOP
1592      0363      77 . .      MOV M,A       ;NO - SET NEW KEYBOARD CHAR
1593      0364      4F . .      MOV C,A
1594      0365      C3 47 03      JMP WTL260     ;PERFORM THE DESIRED FUNCTON
1595      0368      . . .      ;
1596      0368      . . .      WTL270 EQU $
1597      0368      36 E4 .      MVI M,ESCLWD  ;SET <ESC>-<LOWER CASE D> AS
1598      036A      0E 26 .      MVI C,AMPSND  ;CURRENT KEYBOARD CHARACTE
1599      036C      C3 47 03      JMP WTL260     ;PROCESS AMPERSAND
1600      036F      . . .      ;
1601      036F      . . .      WTL280 EQU $
1602      036F      35 . .      DCK M         ;SET TO NEXT STEP CODE
1603      0370      0E 29 .      MVI C,ARPARN  ;ENTER RIGHT PARENTHESIS
1604      0372      C3 47 03      JMP WTL260     ;PROCESS RIGHT PARENTHESIS
1605      0375      . . .      ;
1606      0375      . . .      WTL290 EQU $
1607      0375      36 8E .      MVI M,ESCSO   ;SET <ESC>-<SO> AS CURRENT
1608      0377      0E 43 .      MVI C,C       ;KEYBOARD CHARACTER
1609      0379      C3 47 03      JMP WTL260     ;PROCESS LETTER <C>
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
1611     037C      . . .      ;
1612     037C      . . .      ; DISPLAYABLE CHARACTER - CHECK FOR APPROACHING
1613     037C      . . .      ; END OF LINE WARNING
1614     037C      . . .      ;
1615     037C      . . .      WTL300 EQU $
1616     037C      FE 20 .      CPI CTLLIM      ;CONTROL CODE?
1617     037E      DA A5 03     JC WTL310      ;YES - DON'T LOOK FOR BELL
1618     0381      3A 6E FF     LDA DFLGS      ;NO - GET DATA TRANSFER FLAG
1619     0384      E6 10 .      ANI FCTK2D     ;PROCESSING FUNCTION KEY OR
1620     0386      CC CF 1A     CZ CHKFMS      ;FORMAT/SOFT KEY MODE?
1621     0389      C2 A5 03     JNZ WTL310     ;YES - SKIP BELL COLUMN CHEC
1622     038C      . . .      ;!!!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!!!*
1623     038C      CD 23 60     CALL ZMUCHK    ;AUTOLOT MENU UN?
1624     038F      C2 A5 03     JNZ WTL310     ;YES, IGNORE CHECK
1625     0392      . . .      ;*****
1626     0392      3A D1 FF     LDA ESCFLG     ;NO - GET ESCAPE SEQ FLAG
1627     0395      B7 . .      URA A          ;CURRENTLY IN ESCAPE SEQ?
1628     0396      C2 A5 03     JNZ WTL310     ;YES - DON'T LOOK FOR BELL
1629     0399      3A C1 FF     LDA CURCOL     ;NO - GET CURRENT COLUMN
1630     039C      21 BE FF     LXI H,RHTMGN  ;COMPARE TO RIGHT MARGIN
1631     039F      96 . .      SUB M          ;CLOSE ENOUGH TO RIGHT MARGI
1632     03A0      C6 08 .      ADI BELLIM     ;TO SOUND BELL?
1633     03A2      CC 14 48     CZ ZBELL       ;YES - SOUND BELL
1634     03A5      . . .      ;*****
1635     03A5      . . .      ; PROCESS THE KEY FUNCTION *
1636     03A5      . . .      ;*****
1637     03A5      . . .      WTL310 EQU $
1638     03A5      CD CF 06     CALL LOCLIN    ;PERFORM LOCAL INPUT ROUTINE
1639     03A8      3A 9C FF     LDA CHARIN     ;RECALL KEYBOARD INPUT CHAR
1640     03AB      FE 0D .      CPI CR         ;WAS IT A RETURN?
1641     03AD      C2 94 02     JNZ WTLOOP     ;NO - RESTART WAIT LOOP
1642     03B0      3A F3 FF     LDA MDFLG2     ;YES - GET MODE FLAGS
1643     03B3      E6 04 .      ANI AUTOLF     ;AUTO LINE FEED ENABLED?
1644     03B5      CA 94 02     JZ WTLOOP      ;NO - RESTART WAIT LOOP
1645     03B8      2E 01 .      MVI L,1        ;YES - DELAY 10 MILLISECONDS
1646     03BA      CD E4 13     CALL DELAY     ;THEN SEND LINE FEED
1647     03BD      3E 0A .      MVI A,LF
1648     03BF      C3 D0 02     JMP WTL200     ;FAKE LINE FEED FROM KEYBUAR
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1650	03C2	.	.	;*****	48
1651	03C2	.	.	; S U B R O U T I N E S *	
1652	03C2	.	.	;*****	
1653	03C2	.	.	;	
1654	03C2	.	.	; * * * * *	
1655	03C2	.	.	;	
1656	03C2	.	.	; CHINT - INTERPRET INPUT CHARACTER	
1657	03C2	.	.	;	
1658	03C2	.	.	; ENTRY: C = INPUT CHARACTER	
1659	03C2	.	.	;	
1660	03C2	.	.	; EXIT : Z - FAST STORE USED	
1661	03C2	.	.	; NZ - FULL PROCESSING USED	
1662	03C2	.	.	; A-E,L DESTROYED	
1663	03C2	.	.	;	
1664	03C2	.	.	; TRY FAST STORE FIRST	
1665	03C2	.	.	;	
1666	03C2	.	.	CHINT0 EQU \$;ENTRY FOR I/O INPUT	
1667	03C2	21	6F FF	LXI H,MFLGS2 ;SET H,L TO MODE FLAGS 2	
1668	03C5	79	.	MOV A,C ;PUT INPUT CHAR IN A-REG	
1669	03C6	FE	0A	CPI LF ;CHARACTER = LINE FEED?	
1670	03C8	C2	D4 03	JNZ CHI000 ;NO - CHECK FOR CR/DC3	
1671	03CB	7E	.	MOV A,M ;YES - GET MODE FLAGS 2	
1672	03CC	F6	40	ORI WRPFLG ;TURN ON WRAP FLAG	
1673	03CE	8E	.	CMP M ;WRAP FLAG ALREADY ON?	
1674	03CF	CA	E2 03	JZ CHINT ;YES - EXECUTE LINE FEED	
1675	03D2	77	.	MOV M,A ;NO - SET WRAP FLAG	
1676	03D3	C9	.	RET ;AND IGNORE LINE FEED	
1677	03D4	.	.	;	
1678	03D4	.	.	CHI000 EQU \$	
1679	03D4	FE	0D	CPI CR ;CHARACTER = RETURN?	
1680	03D6	CA	57 04	JZ CHINT1 ;YES - DON'T SET WRAP FLAG	
1681	03D9	FE	13	CPI DC3 ;CHARACTER = DC3?	
1682	03DB	CA	E2 03	JZ CHINT ;YES - DON'T SET WRAP FLAG	
1683	03DE	7E	.	MOV A,M ;NO - SET WRAP FLAG	
1684	03DF	F6	40	ORI WRPFLG	
1685	03E1	77	.	MOV M,A ;UPDATE MODE FLAGS 2	
1686	03E2	.	.	CHINT EQU \$	
1687	03E2	21	67 FF	LXI H,CRAFLG	
1688	03E5	46	.	MOV B,M ;WAS LAST CHARACTER FUNCTION	
1689	03E6	05	.	DCR B ;A CURSOR ADVANCE?	
1690	03E7	FA	57 04	JM CHI100 ;NO - DO FULL PROCESSING	
1691	03EA	70	.	MOV M,B ;YES - CLEAR FLAG	
1692	03EB	79	.	MOV A,C ;PUT INPUT CHARACTER IN A-RE	
1693	03EC	FE	20	CPI CTLLIM ;IS CHARACTER A CONTROL CODE	
1694	03EE	FA	46 04	JM CHI050 ;YES - CHECK FOR DISPLAY FCT	
1695	03F1	.	.	; NO - DO FAST STORE	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1697	03F1	.	.	;	49
1698	03F1	.	.	; FAST STORE PROCESSING	
1699	03F1	.	.	;	
1700	03F1	.	.	CHI010 EQU \$	
1701	03F1	.	.	*****	
1702	03F1	32	89	FF STA DCHAR ;SAVE CHAR FOR POSSIBLE G TX	
1703	03F4	3A	97	90 LDA ZGFLG6 ;IN GRAPHICS TEXT?	
1704	03F7	E6	82	. ANI GTEXT+LABEL	
1705	03F9	C4	4D	60 CNZ ZDPTST ;YES, PUT INTO GRAPHICS	
1706	03FC	D8	.	. RC ;AND DONT PROCESS FURTHER	
1707	03FD	.	.	*****	
1708	03FD	2E	6C	. MVI L,SPOWL-BASE	
1709	03FF	46	.	. MOV B,M ;GET THE SPOW LATCH IN B-REG	
1710	0400	2A	C3	FF LHLD CURADR ;GET LAST CHAR DONE ADDRESS	
1711	0403	7E	.	. MOV A,M ;GET LAST CHARACTER DONE	
1712	0404	B7	.	. ORA A ;IS IT ASCII?	
1713	0405	FA	57	04 JM CHI100 ;NO - DU FULL PROCESSING	
1714	0408	2B	.	. DCX H ;YES - GET NEXT CHARACTER	
1715	0409	7E	.	. MOV A,M	
1716	040A	B7	.	. ORA A ;IS IT ASCII?	
1717	040B	F2	2B	04 JP CHI020 ;YES - OVERLAY EXISTING CHAR	
1718	040E	FE	CC	. CPI EOL ;IS IT EOL?	
1719	0410	C2	57	04 JNZ CHI100 ;NO - DU FULL PROCESSING	
1720	0413	47	.	. MOV B,A ;YES - SAVE EOL AND CLEAR	
1721	0414	2B	.	. DCX H ;SPOW LATCH COMPARE	
1722	0415	7E	.	. MOV A,M ;GET NEXT CHARACTER	
1723	0416	FE	C3	. CPI FILL ;IS IT AN END OF LINE FILL?	
1724	0418	C2	57	04 JNZ CHI100 ;NO - DU FULL PROCESSING	
1725	041B	3A	C0	FF LDA CURROW ;YES - ADD CHAR TO DISPLAY	
1726	041E	F6	40	. ORI MAYEOL ;SET DMA OFF WITH EOL SKIP	
1727	0420	F3	.	. DI ;DISABLE INTERRUPTS	
1728	0421	.	.	*****	
1729	0421	CD	0B	60 CALL ZANCHK ;TURN OFF DMA	
1730	0424	.	.	*****	
1731	0424	3E	04	. MVI A,RSTOFF ;DISABLE RESET KEY	
1732	0426	32	80	83 STA IOKBCU	
1733	0429	70	.	. MOV M,B ;STORE NEW EOL	
1734	042A	23	.	. INX H ;SET TO OLD EOL ADDRESS	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE	50
1736	042B	.	.	.	;	
1737	042B	.	.	.	;	ADD CHARACTER TO DISPLAY
1738	042B	.	.	.	;	
1739	042B	.	.	.	CHI020 EQU \$	
1740	042B	79	.	.	MOV A,C	;RECALL THE INPUT CHARACTER
1741	042C	B8	.	.	CMP B	;STORE INHIBITED BY SPOW?
1742	042D	CA	31	04	JZ CHI030	;YES - BYPASS STORE
1743	0430	71	.	.	MOV M,C	;NO - STORE THE BYTE
1744	0431	.	.	.	CHI030 EQU \$	
1745	0431	CD	D3	10	CALL DISLN1	;TURN DISPLAY BACK ON
1746	0434	22	C3	FF	SHLD CURADR	;STORE NEW CURRENT ADDRESS
1747	0437	21	C8	FF	LXI H,LSTCOL	;INCREMENT LSTCOL
1748	043A	34	.	.	INR M	
1749	043B	CD	89	21	CALL CURADV	;ADVANCE CURSOR
1750	043E	.	.	.	;	*****
1751	043E	.	.	.	;	CHINT2 - SET CURSOR COLUMN ON DISPLAY *
1752	043E	.	.	.	;	*****
1753	043E	.	.	.	;	
1754	043E	.	.	.	;	EXIT : Z TRUE
1755	043E	.	.	.	;	A DESTROYED
1756	043E	.	.	.	;	
1757	043E	.	.	.	CHINT2 EQU \$	
1758	043E	BF	.	.	CMP A	
1759	043F	3A	C1	FF	LDA CURCOL	;GET CURRENT COLUMN NUMBER
1760	0442	32	00	87	STA IUCRCL	;SET DISPLAY CURSOR COLUMN
1761	0445	C9	.	.	RET	;RETURN
1762	0446	.	.	.	;	
1763	0446	.	.	.	;	CONTROL CODE - CHECK FOR DISPLAY FUNCTIONS
1764	0446	.	.	.	;	
1765	0446	.	.	.	CHI050 EQU \$	
1766	0446	CD	72	11	CALL CKDSPF	;DISPLAY FUNCTIONS ENABLED?
1767	0449	CA	57	04	JZ CHI100	;NO - DO FULL PROCESSING
1768	044C	79	.	.	MOV A,C	;YES - RECALL INPUT CHARACTE
1769	044D	FE	0D	.	CPI CR	;IS IT RETURN CHARACTER?
1770	044F	CA	57	04	JZ CHI100	;YES - DO FULL PROCESSING
1771	0452	FE	1B	.	CPI ESC	;IT IT AN ESCAPE?
1772	0454	C2	F1	03	JNZ CHI010	;NO - DO FAST PROCESSING
1773	0457	.	.	.	;	YES - DO FULL PROCESSING

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1775	0457	.	.	;	51
1776	0457	.	.	; FULL PROCESSING	
1777	0457	.	.	;	
1778	0457	.	.	CHINT1 EQU \$	
1779	0457	.	.	CHI100 EQU \$	
1780	0457	61	.	MOV H,C	
1781	0458	69	.	MOV L,C ;SET "CHAR" AND "DCHAR" TO	
1782	0459	22	88	SHLD CHAR ;CURRENT CHARACTER	
1783	045C	CD	AF	CALL CRADV1 ;CLEAR CURADV FLAG	
1784	045F	.	.	*****	
1785	045F	.	.	; DETERMINE CHARACTER FUNCTION *	
1786	045F	.	.	*****	
1787	045F	2A	D2	LHLD RNGTA ;GET CURRENT RANGE TABLE ADD	
1788	0462	.	.	*****	
1789	0462	.	.	; ADVANCE TO NEXT TABLE ENTRY *	
1790	0462	.	.	*****	
1791	0462	.	.	CHI110 EQU \$	
1792	0462	23	.	INX H	
1793	0463	23	.	INX H	
1794	0464	23	.	INX H	
1795	0465	.	.	*****	
1796	0465	.	.	; COMPARE CHARACTER TO CURRENT RANGE *	
1797	0465	.	.	*****	
1798	0465	79	.	MOV A,C ;PUT CHARACTER IN A-REGISTER	
1799	0466	96	.	SUB M ;CHARACTER >= LOWER BOUND?	
1800	0467	23	.	INX H ;(SET H,L TO UPPER BOUND)	
1801	0468	DA	62	JC CHI110 ;NO - ADVANCE TO NEXT ENTRY	
1802	0468	07	.	RLC ;YES - DOUBLE DIFFERENCE AND	
1803	046C	47	.	MOV B,A ;SAVE VALUE IN B-REGISTER	
1804	046D	7E	.	MOV A,M ;GET UPPER BOUND	
1805	046E	.	.	***** GRAPHICS MODIFICATION *****	
1806	046E	E6	7F	ANI 1770 ;DELETE MSB OF UPPER BND	
1807	0470	.	.	*****	
1808	0470	B9	.	CMP C ;CHARACTER <= UPPER BOUND?	
1809	0471	DA	62	JC CHI110 ;NO - ADVANCE TO NEXT ENTRY	
1810	0474	.	.	*****	
1811	0474	.	.	; CHARACTER FUNCTION FOUND - GET FUNCTION ADDR *	
1812	0474	.	.	*****	
1813	0474	.	.	*****	
1814	0474	.	.	; MODIFICATION FOR GRAPHICS	
1815	0474	.	.	; MERGE BIT 8 OF UPPER BOUND WITH MSB OF JUMP	
1816	0474	.	.	; ADDRESS IF NOT INDEX. THIS IS TO ALLOW	
1817	0474	.	.	; ADDRESSES > 32 K	
1818	0474	.	.	*****	
1819	0474	23	.	INX H	
1820	0475	5E	.	MOV E,M ;PUT ADDRESS ENTRY IN	
1821	0476	23	.	INX H ;A (= MSB), E (= LSB)	
1822	0477	7E	.	MOV A,M	
1823	0478	E6	7F	ANI 1770 ;MASK OUT HIGH ORDER BIT	
1824	047A	57	.	MOV D,A ;(PUT NEW MSB INTO D-REG)	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 52
=====
1825     047B      . . .      ;***** GRAPHICS MODIFICATION *****
1826     047B      E5 . .      PUSH H      ;SAVE POINTER TO MSBYTE OF A
1827     047C      2B . .      DCX H      ;GET POINTER TO UPPER BOUND
1828     047D      2B . .      DCX H
1829     047E      7E . .      MOV A,M     ;FETCH UPPER BOUND
1830     047F      E6 80 . .   ANI 200H    ;WANT MSB OF UPPER BOUND
1831     0481      B2 . .      ORA D      ;MERGE WITH TABLE ADDRESS
1832     0482      57 . .      MOV D,A    ;A=NEW MSBYTE OF TABLE ADDRE
1833     0483      E1 . .      POP H      ;RECALL PTR TO REAL MSBYTE
1834     0484      96 . .      SUB M      ;USE INDX TAB IF MSB NOT SET
1835     0485      C2 8E 04    JNZ CHI200 ;DONT USE INDEXTABLE,MSB SET
1836     0488      . . .      ;*****
1837     0488      68 . .      MOV L,B    ;YES - PUT DIFFERENCE IN H,L
1838     0489      67 . .      MOV H,A    ;(A = 0)
1839     048A      19 . .      DAD D      ;COMPUTE TABLE ADDRESS
1840     048B      5E . .      MOV E,M    ;GET INDEX TABLE VALUE
1841     048C      23 . .      INX H
1842     048D      56 . .      MOV D,M
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 53
1844	048E	.	.	*****	
1845	048E	.	.	; PERFORM CHARACTER FUNCTION *	
1846	048E	.	.	*****	
1847	048E	.	.	CHI200 EQU \$	
1848	048E	EB	.	XCHG	
1849	048F	22	CE FF	SHLD CNTFAD ;SET FUNCTION ADDRESS	
1850	0492	06	01 .	MVI B,1 ;SET INITIAL FUNCTION INDEX	
1851	0494	26	FF .	MVI H,BASEH ;SET H TO DATA PAGE	
1852	0496	3E	04 .	MVI A,RSTOFF ;DISABLE RESET KEY	
1853	0498	32	80 83	STA IOKBCU	
1854	049B	.	.	*****	
1855	049B	CD	CD FF	CALL ECONF ;EXECUTE CHARACTER FUNCTION	
1856	049E	.	.	*****	
1857	049E	CD	D9 10	CALL DISLN3 ;RE-ENABLE RESET KEY	
1858	04A1	CD	72 11	CALL CKDSPF ;DISPLAY FUNCTIONS ENABLED?	
1859	04A4	C2	B3 04	JNZ CHI270 ;YES - DON'T END ESCAPE SEQ'	
1860	04A7	21	D1 FF	LXI H,ESCFLG ;NO - CHECK ESCAPE FLAG	
1861	04AA	46	.	MOV B,M	
1862	04AB	05	.	DCR B ;ESCAPE SEQUENCE IN PROGRESS	
1863	04AC	FA	B3 04	JM CHI270 ;NO - DON'T CHANGE ESC FLAG	
1864	04AF	70	.	MOV M,B ;YES - UPDATE ESCAPE COUNTER	
1865	04B0	CC	48 05	CZ ESCEND ;RESET RANGE TABLE POINTER	
1866	04B3	.	.	;	
1867	04B3	.	.	CHI270 EQU \$	
1868	04B3	3A	88 FF	LDA CHAR ;SAVE THE LAST CHARACTER	
1869	04B6	32	69 FF	STA LCHAR ;PROCESSED	
1870	04B9	BC	.	CMP H ;SET Z FALSE	
1871	04BA	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  54
=====
1873     04BB      . . .      ;*****
1874     04BB      . . .      ; CHECK CONTROL CODES FOR BLOCK TERMINATOR OR *
1875     04BB      . . .      ;   BLOCK TRANSFER TRIGGER                       *
1876     04BB      . . .      ;*****
1877     04BB      . . .      ;
1878     04BB      . . .      ; ENTRY:  C = INPUT CHARACTER
1879     04BB      . . .      ;
1880     04BB      . . .      CHKCTL EQU $
1881     04BB      3A 04 50    LDA  BLKTRM   ;GET BLOCK TERMINATOR CHAR
1882     04BE      B9 . .     CMP  C        ;INPUT = BLOCK TERMINATOR?
1883     04BF      CA C6 0D    JZ   SFKYDS   ;YES - DISPLAY INPUT
1884     04C2      3A 6E FF    LDA  DFLGS    ;GET TRANSFER FLAGS
1885     04C5      E6 01 .     ANI  SDACOM   ;INPUT FROM DATA COMM?
1886     04C7      C8 . .     RZ          ;NO - DO NOTHING
1887     04C8      3A 02 50    LDA  TRIGGR   ;IS INPUT CHARACTER THE
1888     04CB      B9 . .     CMP  C        ;BLOCK TRANSFER TRIGGER?
1889     04CC      C0 . .     RNZ         ;NO - DO NOTHING
1890     04CD      . . .      ;
1891     04CD      . . .      CHKCT1 EQU $ ;SET BLOCK TRANSFER TRIGGER
1892     04CD      3E 01 .     MVI  A,SETTRG ;GO TO DATA COMM ROUTINE TO
1893     04CF      C3 73 13    JMP  DCMCTL   ;SET BLOCK TRANSFER TRIGGE
=====

```

ITEM	LOC	OBJCT CODE	SOURCE STATEMENTS	PAGE 55
1895	04D2	. . .	;*****	
1896	04D2	. . .	; DSPTCH - DISPATCH PENDING BLOCK TRANSFERS *	
1897	04D2	. . .	;*****	
1898	04D2	. . .	DSPTCH EQU \$	
1899	04D2	3A F8 FF	LDA CMFLGS ;GET COMMON FLAGS	
1900	04D5	E6 01 .	ANI BLKTRG ;BLOCK TRANSFER TRIGGER SET?	
1901	04D7	C8 . .	RZ ;NO - RETURN	
1902	04D8	3A 70 FF	LDA MFLGS ;YES - RELEASE ANY PENDING	
1903	04DB	21 01 05	LXI H,DSPTAB ;BLOCK TRANSFERS	
1904	04DE	0E 08 .	MVI C,NMPNDG	
1905	04E0	. . .	;	
1906	04E0	. . .	DSP010 EQU \$	
1907	04E0	0F . .	RRC ;TRANSFER PENDING BIT SET?	
1908	04E1	DA FD 04	JC DSP020 ;YES - GO DO TRANSFER	
1909	04E4	23 . .	INX H ;NO - CHECK NEXT BIT	
1910	04E5	23 . .	INX H ;INCREMENT FUNCTION TABLE AD	
1911	04E6	0D . .	DCR C ;ALL BITS CHECKED?	
1912	04E7	C2 E0 04	JNZ DSP010 ;NO - CONTINUE CHECKING	
1913	04EA	. . .	;	
1914	04EA	3A 6F FF	LDA MFLGS2 ;YES - CHECK 2ND SET OF FLAG	
1915	04ED	0F . .	RRC ;DEVICE RECORD PENDING?	
1916	04EE	DA 23 28	JC IORDGO ;YES - SEND I/O RECORD	
1917	04F1	0F . .	RRC ;BINARY DATA PENDING?	
1918	04F2	DA 29 28	JC BNRYGO ;YES - TRANSMIT THE DATA	
1919	04F5	. . .	;*****	
1920	04F5	3A 6B 90	LDA ZGSBLK ;LOAD GRAFIX STATUS BLOCK #	
1921	04F8	B7 . .	ORA A ;ANY GRAPHICS STATUS PENDING	
1922	04F9	C2 5E 60	JNZ ZGSTAT ;YES, SEND PROPER BLOCK	
1923	04FC	. . .	;*****	
1924	04FC	C9 . .	RET ;NO - RETURN	
1925	04FD	. . .	;*****	
1926	04FD	. . .	; PENDING BIT FOUND - GO TO TRANSMIT FUNCTION *	
1927	04FD	. . .	;*****	
1928	04FD	. . .	DSP020 EQU \$	
1929	04FD	CD C6 1A	CALL CHAIN ;GET TRANSMIT FUNCTION ADDR	
1930	0500	E9 . .	PCHL ;GO TO THE FUNCTION	
1931	0501	. . .	;	
1932	0501	. . .	DSPTAB EQU \$	
1933	0501	59 13 .	DW DC2GO ;SEND DC2	
1934	0503	0B 0E .	DW STATGO ;SEND TERMINAL STATUS	
1935	0505	AD 7D .	DW STA2GO ;SEND TERMINAL STATUS 2	
1936	0507	1D 28 .	DW IOSTGO ;SEND I/O STATUS	
1937	0509	15 13 .	DW CRSNGO ;SEND CURSOR ADDRESS	
1938	050B	95 15 .	DW FKEYGO ;SEND FUNCTION KEY DATA	
1939	050D	76 14 .	DW DSPGO ;SEND DISPLAY DATA	
1940	050F	20 28 .	DW IODNGO ;SEND I/O TERMINATION CODE	
1941	0511	. . .	;	
1942	0008	. . .	NMPNDG EQU \$-DSPTAB/2	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1944	0511	.	.	*****	
1945	0511	.	.	; ESCAPE CHARACTER PROCESSING *	
1946	0511	.	.	*****	
1947	0511	.	.	ESCAPE EQU \$	
1948	0511	.	.	*****	
1949	0511	CD	3E 60	CALL ZTKCLR ;CLEAR ECHO SUPPRESS	
1950	0514	.	.	*****	
1951	0514	3A	6E FF	LDA DFLGS	
1952	0517	E6	01 .	ANI SDACOM ;DATA FROM DATACOM?	
1953	0519	CA	26 05	JZ ESC010 ;NO - DON'T LOCK KEYBOARD	
1954	051C	3A	F3 FF	LDA MDFLG2 ;YES - GET MODE FLAGS	
1955	051F	E6	02 .	ANI BLKMODE ;BLOCK MODE?	
1956	0521	3E	08 .	MVI A,ESCINP ;(PUT IGNORE FLAG IN A-REG	
1957	0523	C4	94 18	CNZ SETMF2 ;YES - SET IGNORE KEYBD FLAG	
1958	0526	.	.	ESC010 EQU \$	
1959	0526	CD	E5 1A	CALL CHKSKF ;SOFT KEY MODE?	
1960	0529	21	3D 7E	LXI H,ESCTAB ;(SET FOR NORMAL ESC TABLE	
1961	052C	CA	34 05	JZ ESCAP0 ;NO - SET RANGE TABLE	
1962	052F	21	29 7E	LXI H,SESCTB ;YES - USE SOFT KEY TABLE	
1963	0532	.	.	*****	
1964	0532	.	.	; ESCAP0 - SET RANGE TABLE FOR ESCAPE SEQUENCE *	
1965	0532	.	.	*****	
1966	0532	.	.	;	
1967	0532	.	.	; ENTRY: A = RADIX (BASE) FOR DIGIT PARAMETERS	
1968	0532	.	.	; H,L = ADDRESS OF NEW RANGE TABLE	
1969	0532	.	.	;	
1970	0532	.	.	; EXIT : H,L = ESCFLG	
1971	0532	.	.	;	
1972	0532	.	.	; ESCAPA - USE DECIMAL RADIX	
1973	0532	.	.	;	
1974	0532	.	.	ESCAPA EQU \$	
1975	0532	3E	0A .	MVI A,DECRDX ;SET RADIX FOR BASE 10 DIGIT	
1976	0534	.	.	ESCAP0 EQU \$	
1977	0534	32	D4 FF	STA RADIX ;SET PARAMETER RADIX	
1978	0537	22	D2 FF	SHLD RRGTA ;SET NEW RANGE TABLE	
1979	053A	.	.	ESCAPB EQU \$;ENTRY TO CLEAR ACCUMULATOR	
1980	053A	21	DD FF	LXI H,IOCSGN ;CLEAR OUT THE PARAMETER	
1981	053D	1E	03 .	MVI E,3 ;ACCUMULATOR AREA	
1982	053F	CD	30 12	CALL CLRAL1	
1983	0542	.	.	ESCAP1 EQU \$	
1984	0542	21	D1 FF	LXI H,ESCFLG ;SET FLAG TO RESET AFTER	
1985	0545	36	02 .	MVI M,2 ;FOLLOWING CHARACTER	
1986	0547	C9	.	RET ;RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1988	0548	.	.	*****	57
1989	0548	.	.	; ESCEND - END OF ESCAPE SEQUENCE PROCESSING *	
1990	0548	.	.	*****	
1991	0548	.	.	ESCEND EQU \$	
1992	0548	.	.	*****	
1993	0548	2A	5C 60	LHLD ZMUTB ;ASSUME AUTOPLLOT MENU UP	
1994	054B	CD	23 60	CALL ZMUCHK ;IS IT?	
1995	054E	C2	5D 05	JNZ ESCEN1 ;JUMP IF YES	
1996	0551	.	.	*****	
1997	0551	21	FF 7D	LXI H,RTABLE ;SET FOR NORMAL RANGE TABLE	
1998	0554	CD	E5 1A	CALL CHKSFK ;SOFT KEY MODE?	
1999	0557	CA	5D 05	JZ ESCEN1 ;NO - USE NORMAL TABLE	
2000	055A	21	FB 7D	LXI H,DFSTB0 ;YES - USE SOFT KEY TABLE	
2001	055D	.	.	ESCEN1 EQU \$	
2002	055D	22	D2 FF	SHLD RNGTA ;RESET RANGE TABLE POINTER	
2003	0560	AF	.	XRA A ;CLEAR ESCAPE FLAG AND	
2004	0561	32	D1 FF	STA ESCFLG ;ESCAPE KEYBOARD LOCKOUT	
2005	0564	3E	F7 .	MVI A,377Q-ESCINP ;FLAG	
2006	0566	.	.	*****	
2007	0566	.	.	; CLRMF2 - CLEAR FLAG BIT IN MFLGS2 *	
2008	0566	.	.	*****	
2009	0566	.	.	;	
2010	0566	.	.	; ENTRY: A = 377B - FLAG BIT TO BE CLEARED	
2011	0566	.	.	;	
2012	0566	.	.	; EXIT : A = UPDATED MFLGS2 VALUE	
2013	0566	.	.	; H,L = MFLGS2	
2014	0566	.	.	;	
2015	0566	.	.	CLRMF2 EQU \$	
2016	0566	21	6F FF	LXI H,MFLGS2	
2017	0569	A6	.	ANA M ;CLEAR THE FLAG BIT	
2018	056A	77	.	MOV M,A ;STORE NEW SETTINGS	
2019	056B	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 58
=====
2021      056C      . . . ;
2022      056C      . . . ; * * * * *
2023      056C      . . . ;
2024      056C      . . . ; FNDRAM - LOCATE END OF RAM SPACE
2025      056C      . . . ;
2026      056C      . . . ; ENTRY: B = MSB OF RAM SPACE LOWER LIMIT
2027      056C      . . . ; H,L = ADDR OF UPPER BOUNDARY
2028      056C      . . . ;
2029      056C      . . . ; EXIT : H,L = ADDRESS OF LOWER BOUNDARY
2030      056C      . . . ; A DESTROYED
2031      056C      . . . ;
2032      056C      . . . FNDRAM EQU $
2033      056C      AF . . . XRA A
2034      056D      6F . . . MOV L,A ;SET ADDRESS'S LSB TO ZERO
2035      056E      77 . . . MOV M,A ;SET RAM LOCATION TO ZERO
2036      056F      BE . . . CMP M ;ALL ZEROES STORED?
2037      0570      C2 80 05 JNZ FRM010 ;NO - RAM LIMIT FOUND
2038      0573      35 . . . DCR M ;YES - TRY TO SET TO ALL ONE
2039      0574      34 . . . INR M ;ALL ONES STORED?
2040      0575      C2 80 05 JNZ FRM010 ;NO - RAM LIMIT FOUND
2041      0578      7C . . . MOV A,H ;YES - MOVE TO NEXT 1K
2042      0579      D6 04 . SUI 4
2043      057B      67 . . . MOV H,A
2044      057C      B8 . . . CMP B ;RAM LIMIT REACHED?
2045      057D      F2 6C 05 JP FNDRAM ;NO - TRY NEXT 1K
2046      0580      . . . ;
2047      0580      . . . ; RAM LIMIT FOUND - RETURN LOW BOUNDARY
2048      0580      . . . ;
2049      0580      . . . FRM010 EQU $
2050      0580      7C . . . MOV A,H ;ADJUST H,L TO TRUE LOWER
2051      0581      C6 04 . ADI 4 ;BOUNDARY
2052      0583      E6 FC . ANI 3740 ;MASK FOR 1K START ADDRESS
2053      0585      67 . . . MOV H,A
2054      0586      C9 . . . RET ;RETURN
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2056	0587	.	.	;	59
2057	0587	.	.	; * * * * *	
2058	0587	.	.	;	
2059	0587	.	.	; GETBUF - GET BUFFER SPACE	
2060	0587	.	.	;	
2061	0587	.	.	; ENTRY: B,C = LENGTH OF BUFFER REQUIRED	
2062	0587	.	.	;	
2063	0587	.	.	; EXIT : A,H,L DESTROYED	
2064	0587	.	.	; P - BUFFER SPACE ALLOCATED	
2065	0587	.	.	; D,E = BUFFER START ADDRESS	
2066	0587	.	.	; M - BUFFER SPACE NOT ALLOCATED	
2067	0587	.	.	; D,E DESTROYED	
2068	0587	.	.	;	
2069	0587	.	.	; THIS ROUTINE ALLOCATES A CONTIGUOUS AREA OF	
2070	0587	.	.	; RAM. THE BUFFER SPACE MAY NOT START ON A	
2071	0587	.	.	; 256 BYTE PAGE BOUNDARY.	
2072	0587	.	.	;	
2073	0587	.	.	GETBUF EQU \$	
2074	0587	2A	8B FF	LHLD BUFEND ;GET CURRENT BUFFER END AND	
2075	058A	11	8E FF	LXI D,BUFBGN+1 ;ADDRESS OF BEGIN PTR'S MS	
2076	058D	CD	AE 05	CALL GTB100 ;ENOUGH SPACE?	
2077	0590	FA	99 05	JM GTB010 ;NO - TRY DISPLAY AREA	
2078	0593	22	8B FF	SHLD BUFEND ;YES - STORE NEW BUFFER END	
2079	0596	.	.	GTB005 EQU \$	
2080	0596	EB	.	XCHG ;SET D,E TO LOW ADDRESS	
2081	0597	13	.	INX D ;OF BUFFER	
2082	0598	C9	.	RET ;RETURN	
2083	0599	.	.	;	
2084	0599	.	.	; NOT ENOUGH NON-DISPLAY RAM - TRY DISPLAY AREA	
2085	0599	.	.	;	
2086	0599	.	.	GTB010 EQU \$	
2087	0599	2A	A8 FF	LHLD DSPEND ;GET CURRENT DISPLAY END AND	
2088	059C	11	AB FF	LXI D,DSPBGN+1 ;ADDR OF BEGIN PTR'S MSB	
2089	059F	CD	AE 05	CALL GTB100 ;ENOUGH SPACE?	
2090	05A2	22	A8 FF	SHLD DSPEND ;(STORE NEW DISPLAY END)	
2091	05A5	F2	96 05	JP GTB005 ;YES - RETURN BUFFER ADDRESS	
2092	05A8	21	51 10	LXI H,BUFMSG ;NO - REPORT ERROR	
2093	05AB	C3	85 13	JMP HANGUO	
2094	05AE	.	.	;	
2095	05AE	.	.	; GTB100 - CHECK FOR AVAILABLE SPACE	
2096	05AE	.	.	;	
2097	05AE	.	.	GTB100 EQU \$	
2098	05AE	7D	.	MOV A,L ;SUBTRACT DESIRED SPACE	
2099	05AF	91	.	SUB C ;FROM END OF REGION	
2100	05B0	6F	.	MOV L,A	
2101	05B1	7C	.	MOV A,H	
2102	05B2	98	.	SBB B	
2103	05B3	67	.	MOV H,A	
2104	05B4	EB	.	XCHG ;COMPARE NEW MSB OF END TO	
2105	05B5	BE	.	CMP M ;MSB OF BEGINNING	

13255

13255/90010

2648A MICROCODE LISTING 'PT91'

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                PAGE 60
=====
2106     0586     EB . .      XCHG                          ;PUT NEW END ADDRESS IN H,L
2107     0587     C9 . .      RET                           ;RETURN (P = ENOUGH)
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 61
=====
2109      05B8      . . .      ;*****
2110      05B8      . . .      ; GETDCM - PROCESS DATA COMM INPUT IF ANY *
2111      05B8      . . .      ;*****
2112      05B8      . . .      ;
2113      05B8      . . .      ; ENTRY: DON'T CARE
2114      05B8      . . .      ;
2115      05B8      . . .      ; EXIT : NC
2116      05B8      . . .      ; NZ - DATA COMM INPUT BUFFER EMPTY
2117      05B8      . . .      ; Z - EXIT ON FULL INPUT PROCESSING
2118      05B8      . . .      ; ALL REGISTERS DESTROYED
2119      05B8      . . .      ;
2120      05B8      . . .      GETDCM EQU $
2121      05B8      3A F3 FF      LDA MDFLG2      ;GET HARD MODE FLAGS
2122      05BB      E6 08 .      ANI REMOTE      ;REMOTE MODE ENABLED?
2123      05BD      3A F8 FF      LDA CMFLGS      ;(GET COMMON FLAGS)
2124      05C0      CA 24 06      JZ GDC100      ;NO - IGNORE DATA COMM
2125      05C3      E6 10 .      ANI REMSET      ;WAS REMOTE ON BEFORE?
2126      05C5      CC 26 15      CZ ENTREM      ;NO - SET REMOTE MODE
2127      05C8      . . .      ;*****
2128      05C8      . . .      ; GET DATA COMM INPUT *
2129      05C8      . . .      ;*****
2130      05C8      . . .      GDC010 EQU $
2131      05C8      CD 17 50      CALL ZGETDC     ;ANY DATA COMM INPUT?
2132      05CB      DA 16 06      JC GDC050      ;(PROCESS ERROR IF ANY)
2133      05CE      C0 . .      RNZ             ;NO - RETURN
2134      05CF      4F . .      MOV C,A        ;YES - SAVE INPUT INTO C-REG
2135      05D0      . . .      ;*****
2136      05D0      . . .      ; PROCESS DATA COMM INPUT *
2137      05D0      . . .      ;*****
2138      05D0      . . .      GDC020 EQU $
2139      05D0      . . .      ;*****
2140      05D0      . . .      ; TEST FOR ECHO SUPPRESS ON
2141      05D0      . . .      ; IF SO, DONT PROCESS ANY CHAR UNTIL ONE OF THE
2142      05D0      . . .      ; FOLLOWING IS RECEIVED
2143      05D0      . . .      ; BEL,BS,CR,ESC,GS,HT,LF,RS,US,VT
2144      05D0      3A AD 90      LDA ZTKFLG     ;IS ECHO SUPPRESS ON?
2145      05D3      E6 20 .      ANI SUPCHR
2146      05D5      CA F4 05      JZ GDC027      ;NO, PROCESS NORMALLY
2147      05D8      . . .      ; IF THIS IS THE PROPER CONTROL CODE, CLEAR
2148      05D8      . . .      ; ECHO SUPPRESS AND PROCESS NORMALLY, OTHERWISE,
2149      05D8      . . .      ; IGNORE
2150      05D8      79 . .      MOV A,C        ;RESTORE THE CHAR
2151      05D9      FE 07 .      CPI BEL        ;.LT. BELL?
2152      05DB      DA C8 05      JC GDC010      ;YES, IGNORE
2153      05DE      FE 0E .      CPI CR+1       ;.LT. CR?
2154      05E0      DA ED 05      JC GDC025      ;YES, CLEAR ECHO SUPPRESS
2155      05E3      FE 18 .      CPI ESC        ;.LT. ESCAPE?
2156      05E5      DA C8 05      JC GDC010      ;YES, IGNORE
2157      05E8      FE 20 .      CPI US+1       ;.GT. US??
2158      05EA      D2 C8 05      JNC GDC010     ;YES, IGNORE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 62
2159	05ED	.	.	GDC025 EQU \$	
2160	05ED	.	.	; CLEAR ECHO SUPRESS	
2161	05ED	21	AD 90	LXI H,ZTKFLG	
2162	05F0	3E	DF .	MVI A,-1-SUPCHR	
2163	05F2	A6	.	ANA M	
2164	05F3	77	.	MOV M,A	
2165	05F4	.	.	GDC027 EQU \$	
2166	05F4	.	.	;*****	
2167	05F4	CD	6A 18	CALL SETDF0 ;SET DATA COMM INPUT FLAG	
2168	05F7	3A	F4 FF	LDA MDFLG1 ;GET SOFT MODE FLAGS	
2169	05FA	E6	40 .	ANI RECORD ;RECORD MODE ENABLED?	
2170	05FC	CA	0A 06	JZ GDC030 ;NO - PROCESS THE INPUT	
2171	05FF	79	.	MOV A,C ;YES - LOOK FOR RECORD TRIGG	
2172	0600	FE	0D .	CPI CR ;INPUT = RETURN?	
2173	0602	CA	0A 06	JZ GDC030 ;YES - PROCESS THE CHARACTER	
2174	0605	FE	0A .	CPI LF ;IS IT LINE FEED?	
2175	0607	C2	26 28	JNZ RCRDGO ;NO - EXECUTE RECORD FUNCTIU	
2176	060A	.	.	GDC030 EQU \$;YES - PROCESS THE CHARACTER	
2177	060A	CD	E2 03	CALL CHINT ;PERFORM INPUT PROCEDURE	
2178	060D	CA	C8 05	JZ GDC010 ;FAST STORE - DO SHORT LOOP	
2179	0610	.	.	;	
2180	0610	.	.	GETDC1 EQU \$;SET THE DISPLAY CURSOR	
2181	0610	CD	D3 10	CALL DISLN1 ;SET DISPLAY CURSOR ROW AND	
2182	0613	C3	3E 04	JMP CHINT2 ;COLUMN AND EXIT Z-TRUE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
2184     0616      . . .      ;*****
2185     0616      . . .      ; PROCESS DATA COMM INPUT ERROR *
2186     0616      . . .      ;*****
2187     0616      . . .      GDC050 EQU $
2188     0616      C2 85 13      JNZ HANGUO      ;REPORT AND HANG IF FATAL
2189     0619      CD 72 11      CALL CKDSPF     ;DISPLAY FUNCTIONS ENABLED?
2190     061C      CC 48 05      CZ ESCEND      ;NO - FORCE ESC SEQ ABORT
2191     061F      0E 7F .      MVI C,ADEL     ;FORCE RUBOUT CHARACTER TO
2192     0621      C3 D0 05      JMP GDC020      ;BE DISPLAYED
2193     0624      . . .      ;*****
2194     0624      . . .      ; NOT IN REMOTE MODE - SET TO LOCAL IF NOT *
2195     0624      . . .      ; IN LOCAL MODE ALREADY *
2196     0624      . . .      ;*****
2197     0624      . . .      GDC100 EQU $
2198     0624      E6 10 .      ANI REMSET     ;FIRST TIME IN LOCAL?
2199     0626      C4 0B 15      CNZ ENTLCL     ;YES - SET TO LOCAL MODE
2200     0629      3C . .      INR A          ;FORCE Z FALSE
2201     062A      C9 . .      RET           ;RETURN NO DATA COMM INPUT
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 64
2203	062B	. . .	;*****	
2204	062B	. . .	; GTBLK - GET A NEW DISPLAY BLOCK *	
2205	062B	. . .	;*****	
2206	062B	. . .	;	
2207	062B	. . .	; ENTRY: DON'T CARE	
2208	062B	. . .	;	
2209	062B	. . .	; EXIT : Z - NO BLOCKS AVAILABLE (MEMORY LOCKED)	
2210	062B	. . .	; ALL REGISTERS DESTROYED	
2211	062B	. . .	; NZ - BLOCK ALLOCATED	
2212	062B	. . .	; B,A = H,L = ADDRESS OF CHARACTER	
2213	062B	. . .	; PRECEDING NEXT BLOCK LINK IN BLOCK	
2214	062B	. . .	; C,D,E DESTROYED	
2215	062B	. . .	;	
2216	062B	. . .	GTBLKF EQU \$;GET BLOCK FOR SINGLE CHAR 1	
2217	062B	3E C3 .	MVI A,FILL ;SET FILL CHARACTER TO FILL	
2218	062D	. . .	;	
2219	062D	. . .	GTBLK EQU \$	
2220	062D	32 8F FF	STA FILCHR ;SAVE FILL CHARACTER	
2221	0630	2A AC FF	LHLD FRBLKS ;GET POINTER TO FIRST	
2222	0633	EB . .	XCHG ;FREE BLOCK IN D,E	
2223	0634	7B . .	MOV A,E ;PUT LSB OF LINK IN A-REG	
2224	0635	B7 . .	ORA A ;ANY BLOCKS AVAILABLE?	
2225	0636	CC 10 07	CZ PTBLK ;NO - RELEASE BLOCKS	
2226	0639	CA 10 0C	JZ MLOCK ;AND FORCE MEMORY LOCK ON	
2227	063C	E6 F0 .	ANI 377Q-BLKSM ;COMPUTE ADDRESS OF	
2228	063E	6F . .	MOV L,A ;NEXT BLOCK LINK	
2229	063F	62 . .	MOV H,D	
2230	0640	7E . .	MOV A,M ;GET LSB OF NEXT BLOCK LINK	
2231	0641	4F . .	MOV C,A ;SAVE LSB IN C-REGISTER	
2232	0642	2F . .	CMA ;END OF LINE LINK (LOWER	
2233	0643	E6 0F .	ANI BLKSM ;FOUR BITS # ALL ONES)?	
2234	0645	CA 56 06	JZ GBL100 ;NO - RELEASE NEXT BLOCK	
2235	0648	. . .	;*****	
2236	0648	. . .	; RELEASE LAST BLOCK OF LINE *	
2237	0648	. . .	;*****	
2238	0648	13 . .	INX D ;SET H,L TO LSB PART OF PREV	
2239	0649	13 . .	INX D ;LINE LINK	
2240	064A	6B . .	MOV L,E	
2241	064B	CD C6 1A	CALL CHAIN ;GET PREV LINE ADDR IN H,L	
2242	064E	22 AC FF	SHLD FRBLKS ;SET AS NEW FREE BLOCKS HEAD	
2243	0651	42 . .	MOV B,D ;PUT CURRENT BLOCK ADDRESS	
2244	0652	7B . .	MOV A,E ;IN B,A	
2245	0653	C3 64 06	JMP GBL200 ;FILL BLOCK WITH FILL CHARS	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 65
=====
2247      0656      . . .      ;*****
2248      0656      . . .      ; RELEASE NEXT BLOCK OF LINE *
2249      0656      . . .      ;*****
2250      0656      . . .      GBL100 EQU $
2251      0656      23 . . .      INX H ;GET MSB OF NEXT BLOCK LINK
2252      0657      46 . . .      MOV B,M
2253      0658      28 . . .      DCX H ;RESTORE H,L TO ADDRESS OF
2254      0659      . . .      ; OF LSB PART IN FIRST BLOCK
2255      0659      79 . . .      MOV A,C ;COMPUTE ADDRESS OF NEXT
2256      065A      E6 F0 . . . ANI 377Q-BLKSM ;BLOCK LINK IN SECOND BLOC
2257      065C      4F . . .      MOV C,A
2258      065D      0A . . .      LDAX B ;TRANSFER NEXT BLOCK LINK OF
2259      065E      77 . . .      MOV M,A ;SECOND BLOCK TO NEXT BLOC
2260      065F      03 . . .      INX B ;LINK IN FIRST BLOCK
2261      0660      23 . . .      INX H
2262      0661      0A . . .      LDAX B
2263      0662      77 . . .      MOV M,A
2264      0663      79 . . .      MOV A,C ;SET A-REGISTER FOR "BLNKFL"
2265      0664      . . .      ;*****
2266      0664      . . .      ; FILL BLOCK WITH SPECIFIED FILL CHARACTER *
2267      0664      . . .      ;*****
2268      0664      . . .      ;
2269      0664      . . .      ; B,A = ANY ADDRESS IN BLOCK
2270      0664      . . .      ; FILCHR = CHARACTER TO FILL BLOCK WITH
2271      0664      . . .      ;
2272      0664      . . .      GBL200 EQU $
2273      0664      F6 0F . . . ORI BLKSM ;SET H,L TO ADDRESS OF LAST
2274      0666      6F . . .      MOV L,A ;DISPLAY CHARACTER POSITIO
2275      0667      60 . . .      MOV H,B ;IN BLOCK
2276      0668      0E 0D . . . MVI C,BLKSZ-3 ;SET FILL COUNT
2277      066A      3A 8F FF LDA FILCHR ;GET THE FILL CHARACTER
2278      066D      . . .      GBL210 EQU $
2279      066D      77 . . .      MOV M,A ;STORE THE FILL CHARACTER
2280      066E      2B . . .      DCX H ;MOVE TO NEXT BYTE
2281      066F      0D . . .      DCR C ;BLUCK FILL COMPLETED?
2282      0670      C2 6D 06 JNZ GBL210 ;NO - CONTINUE FILLING
2283      0673      77 . . .      MOV M,A ;YES - WRITE LAST PAD
2284      0674      7D . . .      MOV A,L ;SET B,A TO EXIT ADDRESS
2285      0675      B7 . . .      ORA A ;SET NZ
2286      0676      C9 . . .      RET ;RETURN
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
2288	0677	. . .	;*****	66
2289	0677	. . .	; GTNWLN - START A NEW LINE *	
2290	0677	. . .	;*****	
2291	0677	. . .	;	
2292	0677	. . .	; ENTRY: LLINE = ADDRESS OF PREVIOUS LINE	
2293	0677	. . .	;	
2294	0677	. . .	; EXIT : NZ - NO BLOCKS AVAILABLE (MEMORY LOCK)	
2295	0677	. . .	; ALL REGISTERS DESTROYED	
2296	0677	. . .	; Z - LINE ALLOCATED	
2297	0677	. . .	; H,L = ADDR OF FIRST CHAR IN NEW LINE	
2298	0677	. . .	; LLINE = ADDR OF LSB PART OF NEXT LINE	
2299	0677	. . .	; POINTER IN THE NEW LINE	
2300	0677	. . .	; A-E DESTROYED	
2301	0677	. . .	;	
2302	0677	. . .	; NEW LINE IS LINKED TO PREVIOUS LINE IF PREVIOUS	
2303	0677	. . .	; LINE EXISTS (I.E., LSB OF PREV LINE ADDR # 0)	
2304	0677	. . .	;	
2305	0677	. . .	GTNWLN EQU \$	
2306	0677	3E C0	MVI A,STPR ;SET LAST FORMAT CONTROL COD	
2307	0679	32 C5 FF	STA LSTFMT ;TO START PROTECT	
2308	067C	CD 2B 06	CALL GTBLKF ;GET A BLOCK FROM FREE LIST	
2309	067F	CA 0A 0C	JZ NZEXIT ;RETURN NZ IF NO BLOCKS	
2310	0682	EB . .	XCHG ;D,E = NEW BLOCK ADDRESS	
2311	0683	2A A1 FF	LHLD LLINE ;GET ADDRESS OF PREVIOUS	
2312	0686	EB . .	XCHG ;LINE IN D,E	
2313	0687	F6 0F .	ORI BLKSM ;COMPUTE ADDRESS OF LSB PART	
2314	0689	06 02 .	SUI 2 ;OF NEXT LINE LINK	
2315	068B	2B . .	DCX H ;STORE ADDRESS INTO NEXT	
2316	068C	70 . .	MOV M,B ;BLOCK LINK	
2317	068D	2D . .	DCR L ;(USE DCR TO AVOID CARRY)	
2318	068E	77 . .	MOV M,A	
2319	068F	C6 02 .	ADI 2 ;SET ADDRESS TO MSB PART OF	
2320	0691	6F . .	MOV L,A ;PREVIOUS LINE LINK	
2321	0692	72 . .	MOV M,D ;SET PREVIOUS LINE LINK TO	
2322	0693	2B . .	DCX H ;POINT TO OLD LINE	
2323	0694	73 . .	MOV M,E	
2324	0695	2B . .	DCX H	
2325	0696	36 CE .	MVI M,EOP ;SET NEXT LINE LINK TO "EOP"	
2326	0698	2B . .	DCX H	
2327	0699	AF . .	XRA A ;SET TERMINATOR (LSB = 0)	
2328	069A	77 . .	MOV M,A	
2329	069B	22 A1 FF	SHLD LLINE ;STORE NEW LAST LINE ADDRESS	
2330	069E	2B . .	DCX H	
2331	069F	CD 79 0E	CALL STCHR1 ;SET FIRST DISPLAY CHARACTER	


```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 67
=====
2333     06A2      . . .      ;*****
2334     06A2      . . .      ; LINK NEW LINE BACK TO PREVIOUS LAST LINE *
2335     06A2      . . .      ;*****
2336     06A2      B3 . .      ORA E      ;PREVIOUS LINE EXIST (LSB#0)
2337     06A3      C8 . .      RZ        ;NO - RETURN
2338     06A4      EB . .      XCHG      ;YES - LINK NEW LINE TO
2339     06A5      73 . .      MOV M,E   ;PREVIOUS LINE
2340     06A6      23 . .      INX H
2341     06A7      72 . .      MOV M,D
2342     06A8      EB . .      XCHG      ;RESTORE H,L
2343     06A9      BF . .      CMP A     ;SET Z TRUE
2344     06AA      C9 . .      RET      ;RETURN
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 68
=====
2346      06AB      . . .      ;*****
2347      06AB      . . .      ; INITDS - SET UP INITIAL DISPLAY VALUES *
2348      06AB      . . .      ;*****
2349      06AB      . . .      ;
2350      06AB      . . .      ; EXIT : H,L = ADDRESS OF THE LSB PART OF THE
2351      06AB      . . .      ;           NEXT LINE POINTER IN THE INITIAL
2352      06AB      . . .      ;           DISPLAY BLOCK
2353      06AB      . . .      ;           A DESTROYED
2354      06AB      . . .      ;
2355      06AB      . . .      ; THIS ROUTINE ALLOCATES THE INITIAL LINE OF
2356      06AB      . . .      ; THE DISPLAY AND INITIALIZES THE DISPLAY
2357      06AB      . . .      ; PARAMETERS:
2358      06AB      . . .      ;
2359      06AB      . . .      ; DISPST,CURADR = ADDRESS OF THE FIRST DISPLAY
2360      06AB      . . .      ; CHARACTER IN THE INITIAL DISPLAY BLOCK
2361      06AB      . . .      ;
2362      06AB      . . .      ; LSTLIN,FLINE,TOPLIN = ADDRESS OF THE LSB
2363      06AB      . . .      ; PART OF THE NEXT LINE POINTER IN THE
2364      06AB      . . .      ; INITIAL DISPLAY BLOCK
2365      06AB      . . .      ;
2366      06AB      . . .      ; RHTMGN = MAXCOL (= 79)
2367      06AB      . . .      ;
2368      06AB      . . .      ;
2369      06AB      CD 77 06      INITDS EQU $
2370      06AE      22 FE FF      CALL GTNWLN      ;GET INITIAL DISPLAY BLOCK
2371      06B1      22 C3 FF      SHLD DISPST      ;SET THE DISPLAY POINTER
2372      06B4      23 . .      SHLD CURADR      ;AND THE CURRENT CHAR ADDR
2373      06B5      22 C9 FF      INX H
2374      06B8      22 9F FF      SHLD LSTLIN      ;SET THE CURRENT LINE
2375      06BB      22 CB FF      SHLD FLINE      ;PARAMETERS
2376      06BE      3E 4F .      SHLD TOPLIN
2377      06C0      32 BE FF      MVI A,MAXCOL    ;INITIALIZE THE RIGHT MARGIN
2378      06C3      C9 . .      STA RHTMGN      ;TO THE LAST COLUMN
                RET                                ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2380	06C4	.	.	*****	69
2381	06C4	.	.	; LOCLIN - PROCESS LOCAL DATA ENTRY *	
2382	06C4	.	.	*****	
2383	06C4	.	.	;	
2384	06C4	.	.	; ENTRY: C = INPUT CHARACTER	
2385	06C4	.	.	; (CHARIN) = KEYBOARD INPUT CODE	
2386	06C4	.	.	;	
2387	06C4	.	.	; EXIT : ALL REGISTERS DESTROYED	
2388	06C4	.	.	;	
2389	06C4	.	.	; THIS ROUTINE PROCESSES INPUT CHARACTERS FROM	
2390	06C4	.	.	; KEYBOARD. THE ROUTINE DETERMINES WHETHER OR	
2391	06C4	.	.	; NOT THE CHARACTER SHOULD BE TRANSMITTED OR	
2392	06C4	.	.	; PROCESSED LOCALLY	
2393	06C4	.	.	;	
2394	06C4	.	.	; LOCLIO - PROCESS FUNCTIONAL KEY INPUT	
2395	06C4	.	.	;	
2396	06C4	.	.	LOCLIO EQU \$	
2397	06C4	3A	FB	LDA KBJMPR ;GET KEYBOARD JUMPERS A-H	
2398	06C7	E6	01	ANI CONDIS ;DISPLAY ALL FUNCTIONS OR	
2399	06C9	CC	72	CZ CKDSPF ;DISPLAY FUNCTIONS ENABLED	
2400	06CC	CA	01	JZ LCI050 ;NO - PROCESS LOCALLY ONLY	
2401	06CF	.	.	*****	
2402	06CF	.	.	; TRANSMIT CODE IF IN REMOTE CHARACTER MODE *	
2403	06CF	.	.	*****	
2404	06CF	.	.	LOCLIN EQU \$	
2405	06CF	.	.	*****	
2406	06CF	3A	96	LDA ZAPFLG ;IS AUTO PLOT ON?	
2407	06D2	E6	02	ANI APIP	
2408	06D4	C4	29	CNZ ZAPCHK ;IF YES, PROCESS FURTHER	
2409	06D7	.	.	*****	
2410	06D7	CD	E5	CALL CHKSKF ;SOFT KEY DEFINE MODE?	
2411	06DA	C2	01	JNZ LCI050 ;YES - PROCESS LOCALLY ONLY	
2412	06DD	3A	F3	LDA MDFLG2 ;NO - GET HARD MODE FLAGS	
2413	06E0	E6	0A	ANI REMOTE+BLKMDE	
2414	06E2	EE	08	XRI REMOTE ;REMOTE AND NOT BLOCK MODE?	
2415	06E4	C2	01	JNZ LCI050 ;NO - PROCESS LOCALLY ONLY	
2416	06E7	.	.	!!!!!!! GRAPHICS MODIFICATION !!!!!!!!*	
2417	06E7	CD	23	CALL ZMUCHK ;AUTO PLOT MENU ON?	
2418	06EA	C2	01	JNZ LCI050 ;YES, PROCESS LOCAL ONLY	
2419	06ED	21	AD	LXI H,ZTKFLG ;IN TEK GIN MODE?	
2420	06F0	3E	10	MVI A,GINMOD	
2421	06F2	A6	.	ANA M	
2422	06F3	C2	01	JNZ LCI050 ;YES, PROCESS LOCAL ONLY	
2423	06F6	.	.	*****	
2424	06F6	79	.	MOV A,C ;YES - RECALL THE INPUT	
2425	06F7	CD	22	CALL XPUTDC ;OUTPUT THE CHARACTER	
2426	06FA	D8	.	RC ;(RETURN IF OUTPUT ERROR)	
2427	06FB	3A	FC	LDA KBDCSW ;GET THE DATA COMM SWITCHES	
2428	06FE	E6	80	ANI FULDUP ;FULL DUPLEX?	
2429	0700	C0	.	RNZ ;YES - RETURN	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 70
=====
2430      0701      . . .      ;                                NO - PROCESS INPUT LOCALLY
2431      0701      . . .      ;*****
2432      0701      . . .      ; PROCESS THE INPUT LOCALLY *
2433      0701      . . .      ;*****
2434      0701      . . .      ;
2435      0701      . . .      ; INPUT CHARACTER IN C-REGISTER
2436      0701      . . .      ;
2437      0701      . . .      LCI050 EQU $
2438      0701      CD CF 1A    CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE
2439      0704      C2 57 04    JNZ CHINT1 ;YES - FORCE FULL PROCESSING
2440      0707      . . .      ;!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!*
2441      0707      CD 23 60    CALL ZMUCHK ;AUTO PLOT MENU ON?
2442      070A      C2 57 04    JNZ CHINT1 ;YES, DO FULL PROCESS
2443      070D      . . .      ;*****
2444      070D      C3 E2 03    JMP CHINT ;NO - 1ST TRY FAST PROCESSIN
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 71
2446	0710	. . .	;*****	
2447	0710	. . .	; PTBLK - RELEASE A LINE TO THE FREE LIST FROM *	
2448	0710	. . .	; THE DISPLAY LIST *	
2449	0710	. . .	;*****	
2450	0710	. . .	;	
2451	0710	. . .	; ENTRY: DON'T CARE	
2452	0710	. . .	;	
2453	0710	. . .	; EXIT : Z - LINE NOT RELEASED	
2454	0710	. . .	; NC - MEMORY LOCKED	
2455	0710	. . .	; C - OUTPUT FAILED FOR EDIT MODE	
2456	0710	. . .	; ALL REGISTERS DESTROYED	
2457	0710	. . .	; NZ - LINE RELEASED	
2458	0710	. . .	; D,E = ADDRESS OF FIFTH BYTE FROM	
2459	0710	. . .	; A = E	
2460	0710	. . .	; B,C,H,L DESTROYED	
2461	0710	. . .	;	
2462	0710	. . .	PTBLK EQU \$	
2463	0710	CD DA 1A	CALL CHKMLK ;MEMORY LOCK ENABLED?	
2464	0713	CA 10 0C	JZ MLOCK ;YES - FLAG MEMORY FULL	
2465	0716	CD 44 07	CALL PTB100 ;SWITCH DISPLAY PARAMETERS	
2466	0719	. . .	; IF IN SOFT KEY MODE	
2467	0719	3A 9A FF	LDA NROWS ;GET NUMBER OF ROWS NEEDED	
2468	071C	B7 . .	ORA A ;NEW ROWS BEING ADDED?	
2469	071D	CC 78 11	CZ CKEDIT ;NO - EDIT MODE?	
2470	0720	C2 4E 07	JNZ PTR200 ;YES - RELEASE TOP LINE	
2471	0723	2A C9 FF	LHLD LSTLIN ;NO - GET CURRENT LINE ADDR	
2472	0726	B6 . .	ORA M ;CURRENTLY IN THE LAST LINE	
2473	0727	CA 4E 07	JZ PTB200 ;YES - RELEASE TOP LINE	
2474	072A	. . .	; NO - RELEASE BOTTOM LINE	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2476	072A	.	.	*****	72
2477	072A	.	.	; RELEASE LAST LINE OF MEMORY *	
2478	072A	.	.	; UPDATE LAST LINE POINTER *	
2479	072A	.	.	*****	
2480	072A	2A	A1 FF	LHLD LLINE ;GET LAST LINE ADDRESS	
2481	072D	23	.	INX H ;GET PREVIOUS LINE ADDRESS	
2482	072E	23	.	INX H	
2483	072F	5E	.	MOV E,M	
2484	0730	23	.	INX H	
2485	0731	56	.	MOV D,M	
2486	0732	EB	.	XCHG	
2487	0733	22	A1 FF	SHLD LLINE ;SET PREV LINE AS LAST LINE	
2488	0736	.	.	*****	
2489	0736	.	.	; STORE EOP IN NEW LAST LINE *	
2490	0736	.	.	*****	
2491	0736	36	00 .	MVI M,0 ;SET TERMINATOR CODE IN	
2492	0738	23	.	INX H ;NEW LAST LINE	
2493	0739	36	CE .	MVI M,EOP	
2494	073B	1B	.	DCX D ;SET D,E TO POINT TO LSB PAR	
2495	073C	1B	.	DCX D ;NEXT LINE POINTER IN OLD	
2496	073D	1B	.	DCX D ;LAST LINE	
2497	073E	C3	89 07	JMP PTB300 ;ADD LINE TO FREE LIST	
2498	0741	.	.	*****	
2499	0741	.	.	; PTB100 - SET PROPER DISPLAY PARAMETERS *	
2500	0741	.	.	*****	
2501	0741	.	.	PTB090 EQU \$;I/O OUTPUT FAIL EXIT	
2502	0741	CD	22 0C	CALL MLK010 ;CLEAR ROWS ALLOCATED FLAG	
2503	0744	.	.	PTB100 EQU \$	
2504	0744	CD	E5 1A	CALL CHKSFK ;SOFT KEY DEFINE MODE?	
2505	0747	37	.	STC ;(SET C-FLAG FOR I/O FAIL)	
2506	0748	C2	1D 23	JNZ SWAP1 ;YES - SWAP DISPLAY PARMS	
2507	074B	C9	.	RET ;NO - RETURN	
2508	074C	00	.	NOP ;"NOP'S" FOR PATCH TO "PT772"	
2509	074D	00	.	NOP	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 73
2511	074E	.	.	*****	
2512	074E	.	.	; RELEASE FIRST LINE OF MEMORY *	
2513	074E	.	.	*****	
2514	074E	.	.	PTB200 EQU \$	
2515	074E	2A	CB FF	LHLD TOPLIN ;GET TOP LINE ADDRESS	
2516	0751	23	.	INX H ;SET FOR PREVIOUS LINE	
2517	0752	23	.	INX H ;ADDRESS	
2518	0753	7E	.	MOV A,M	
2519	0754	B7	.	ORA A ;TOP LINE = FIRST LINE?	
2520	0755	C2	6E 07	JNZ PTB220 ;FIRST LINE IS NOT TOP LINE	
2521	0758	.	.	*****	
2522	0758	.	.	; TOP LINE OF DISPLAY IS FIRST LINE OF MEMORY *	
2523	0758	.	.	; DO ROLL-UP *	
2524	0758	.	.	*****	
2525	0758	21	C0 FF	LXI H,CURROW	
2526	075B	3A	6B FF	LDA MLKROW ;USER WORKING IN FIRST	
2527	075E	BE	.	CMP M ;UNLOCKED ROW?	
2528	075F	C4	30 0D	CNZ ROLLUP ;NO - ROLL UP DISPLAY	
2529	0762	CA	0D 0C	JZ MLOCK0 ;ROLL UP FAIL - LOCK MEMORY	
2530	0765	21	C0 FF	LXI H,CURROW ;DECREMENT CURSOR ROW	
2531	0768	46	.	MOV B,M	
2532	0769	05	.	DCR B	
2533	076A	FA	6E 07	JM PTB220 ;DON'T STORE IF ROW = 0	
2534	076D	70	.	MOV M,B	
2535	076E	.	.	*****	
2536	076E	.	.	; ADVANCE FIRST LINE POINTER *	
2537	076E	.	.	*****	
2538	076E	.	.	PTB220 EQU \$	
2539	076E	2A	9F FF	LHLD FLINE ;GET ADDRESS OF FIRST DISPLA	
2540	0771	EB	.	XCHG ;LINE	
2541	0772	CD	78 11	CALL CKREDIT ;EDIT MODE ENABLED?	
2542	0775	C4	32 28	CNZ PTTPLN ;YES - TRY TO OUTPUT LINE	
2543	0778	DA	41 07	JC PTB090 ;OUTPUT FAILED - RETURN FAIL	
2544	077B	EB	.	XCHG ;PUT ADDRESS BACK INTO D,E	
2545	077C	5E	.	MOV E,M ;GET ADDRESS OF NEW FIRST	
2546	077D	23	.	INX H ;FIRST LINE	
2547	077E	56	.	MOV D,M	
2548	077F	13	.	INX D ;SET TO NEXT LINE POINTER	
2549	0780	EB	.	XCHG	
2550	0781	22	9F FF	SHLD FLINE ;STORE AS NEW FIRST LINE	
2551	0784	.	.	*****	
2552	0784	.	.	; CLEAR PREVIOUS LINE PNTR IN NEW FIRST LINE *	
2553	0784	.	.	*****	
2554	0784	23	.	INX H ;ADVANCE TO PREVIOUS LINE	
2555	0785	23	.	INX H ;POINTER	
2556	0786	36	00 .	MVI M,0 ;ZERO LSB TO FLAG AS TOP LIN	
2557	0788	1B	.	DCX D ;SET D,E TO LSB OF NEXT LINE	
2558	0789	.	.	; POINTER IN LINE TO BE	
2559	0789	.	.	; RELEASED	

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 74
=====
2561      0789      . . .      ;*****
2562      0789      . . .      ; RELEASE LINE *
2563      0789      . . .      ; D,E = START ADDRESS OF LINE *
2564      0789      . . .      ;*****
2565      0789      . . .      PTB300 EQU $
2566      0789      D5 . .      PUSH D ;SAVE REGISTERS D,E
2567      078A      CD 44 07      CALL PTB100 ;RESTORE PROPER DISPLAY PARM
2568      078D      D1 . .      POP D ;RESTORE D,E
=====
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 75
2570	078E	.	.	. ;*****	
2571	078E	.	.	. ; PUTLIN - ADD LINE TO FREE LIST *	
2572	078E	.	.	. ;*****	
2573	078E	.	.	. ;	
2574	078E	.	.	. ; ENTRY: D,E = ADDRESS OF NEXT LINE FIELD'S LSB	
2575	078E	.	.	. ; OF LINE TO BE RELEASED	
2576	078E	.	.	. ;	
2577	078E	.	.	. ; EXIT : D,E UNCHANGED	
2578	078E	.	.	. ; A = E	
2579	078E	.	.	. ; Z FALSE	
2580	078E	.	.	. ; H,L DESTROYED	
2581	078E	.	.	. ; FREE BLOCKS LIST UPDATED TO INCLUDE	
2582	078E	.	.	. ; RELEASE LINE	
2583	078E	.	.	. ;	
2584	078E	.	.	. PUTLIN EQU \$	
2585	078E	CD	F1 11	CALL MLKOF ;RESET MEMORY LOCKED FLAG	
2586	0791	2A	AC FF	LHLD FRBLKS ;GET CURRENT FREE BLOCKS HEA	
2587	0794	EB	.	XCHG ;SET H,L TO MSB PART OF NEXT	
2588	0795	22	AC FF	SHLD FRBLKS ;SET FREE BLOCKS POINTER TO	
2589	0798	7D	.	MOV A,L ;RELEASED LINE	
2590	0799	23	.	INX H ;PUT PREVIOUS FREE BLOCKS	
2591	079A	23	.	INX H ;HEAD INTO PREVIOUS LINE	
2592	079B	73	.	MOV M,E ;POINTER OF RELEASED LINE	
2593	079C	2C	.	INR L ;(USE INR TO FORCE NZ)	
2594	079D	72	.	MOV M,D	
2595	079E	EB	.	XCHG ;RELEASED LINE ADDRESS IN D,	
2596	079F	5F	.	MOV E,A ;SET A = E	
2597	07A0	C9	.	RET ;RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 76
2599	07A1	.	.	. ;*****	
2600	07A1	.	.	. ; RCADRA - LOCATE CURRENT CURSOR POSITION *	
2601	07A1	.	.	. ; IF POSITION EXIST - DON'T EXTEND DISPLAY *	
2602	07A1	.	.	. ;*****	
2603	07A1	.	.	. ;	
2604	07A1	.	.	. ; ENTRY: DON'T CARE	
2605	07A1	.	.	. ;	
2606	07A1	.	.	. ; EXIT : SEE "RCADDR"	
2607	07A1	.	.	. ;	
2608	07A1	.	.	. RCADRA EQU \$	
2609	07A1	CD	AF 21	CALL CRADV1 ;CLEAR CURSOR ADVANCE FLAG	
2610	07A4	3E	01 .	MVI A,IGNTRM ;SET TO IGNORE NON-DISPLAYIN	
2611	07A6	32	6D FF	STA TRMECT ;TERMINATOR	
2612	07A9	.	.	. RCADR8 EQU \$	
2613	07A9	3E	FF .	MVI A,3770 ;SET "BLKFIL" TO INHIBIT	
2614	07AB	32	91 FF	STA BLKFIL ;LINE EXTENSION	
2615	07AE	C3	B5 07	JMP RCADR2 ;LOCATE CURSOR POSITION	
2616	07B1	.	.	. ;*****	
2617	07B1	.	.	. ; LOCATE ADDR CORRESPONDING TO ROW/COLUMN *	
2618	07B1	.	.	. ; DO NOT ADD ROWS IF ROW DOES NOT EXIST *	
2619	07B1	.	.	. ;*****	
2620	07B1	.	.	. RCADR1 EQU \$	
2621	07B1	AF	.	XRA A ;SET TO LOCATE COLUMN 0	
2622	07B2	32	C1 FF	STA CURCOL ;IN DESIRED ROW	
2623	07B5	.	.	. RCADR2 EQU \$	
2624	07B5	3A	C1 FF	LDA CURCOL ;GET THE CURRENT COLUMN	
2625	07B8	.	.	. RCADR3 EQU \$	
2626	07B8	21	9A FF	LXI H,NROWS ;SET "NROWS" TO INHIBIT	
2627	07BB	36	FF .	MVI M,3770 ;BUILDING OF NEW ROWS	
2628	07BD	CD	10 08	CALL RCADR0 ;FIND CHARACTER ADDRESS	
2629	07C0	21	9A FF	LXI H,NROWS ;RESET BUILD INHIBIT FLAGS	
2630	07C3	36	00 .	MVI M,0 ;WITHOUT CHANGING PROCESSO	
2631	07C5	2E	91 .	MVI L,BLKFIL-BASE ;FLAGS	
2632	07C7	36	00 .	MVI M,0	
2633	07C9	C9	.	. RET ;RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 77
2635	07CA	.	.	. ;*****	
2636	07CA	.	.	. ; RCADR4 - GET ADDRESS OF FIRST CHARACTER AFTER *	
2637	07CA	.	.	. ; AFTER PREVIOUS ROW AND COLUMN *	
2638	07CA	.	.	. ;*****	
2639	07CA	.	.	. ;	
2640	07CA	.	.	. ; ENTRY: CURROW = CURRENT ROW	
2641	07CA	.	.	. ; CURCOL = CURRENT COLUMN	
2642	07CA	.	.	. ;	
2643	07CA	.	.	. ; EXIT : Z - CHARACTER FOUND	
2644	07CA	.	.	. ; C = COLUMN NUMBER	
2645	07CA	.	.	. ; D,E = CHARACTER ADDRESS	
2646	07CA	.	.	. ; IF FORMAT MODE ENABLED	
2647	07CA	.	.	. ; B = -1, CHARACTER PROTECTED	
2648	07CA	.	.	. ; # -1, CHARACTER NOT PROTECTED	
2649	07CA	.	.	. ; OTHERWISE, B DESTROYED	
2650	07CA	.	.	. ; A,H,L DESTROYED	
2651	07CA	.	.	. ; NZ - CHARACTER NOT FOUND	
2652	07CA	.	.	. ; ALL REGISTERS DESTROYED	
2653	07CA	.	.	. ;	
2654	07CA	.	.	. RCADR4 EQU \$	
2655	07CA	3A	C1	FF LDA CURCOL ;GET CURRENT COLUMN NUMBER	
2656	07CD	3D	.	. DCR A ;SET FOR PREVIOUS COLUMN	
2657	07CE	CD	B8	07 CALL RCADR3 ;DOES CHARACTER EXIST	
2658	07D1	C0	.	. RNZ ;NO - RETURN	
2659	07D2	4F	.	. MOV C,A ;YES - SAVE COLUMN FOUND IN	
2660	07D3	0C	.	. INR C ;ADVANCE TO NEXT COLUMN	
2661	07D4	CD	90	0C CALL NXTCHR ;GET NEXT CHARACTER	
2662	07D7	CD	CF	1A CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE	
2663	07DA	47	.	. MOV B,A ;(SET B TO INDICATE NOT	
2664	07DB	.	.	. ; PROTECTED IF NOT FORMAT)	
2665	07DB	.	.	. ; NEXT STATEMENT RETURNS)	
2666	07DB	C8	.	. RZ ;NO - RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 78
=====
2668      07DC      . . .      ;
2669      07DC      . . .      ; FORMAT MODE - SEE IF NEXT ASCII CHAR PROTECTED
2670      07DC      . . .      ;
2671      07DC      CD 90 11      CALL CKPROT      ;PREVIOUS CHAR PROTECTED?
2672      07DF      CA F4 07      JZ   RCA440      ;YES - SEE IF NEXT CHAR UNPR
2673      07E2      . . .      ;*****
2674      07E2      . . .      ; LAST CHAR WAS UNPROTECTED *
2675      07E2      . . .      ; SEE IF NEXT CHAR IS PROTECTED *
2676      07E2      . . .      ;*****
2677      07E2      CD 1B 20      CALL FNDCH0      ;IS NEXT CHARACTER PROTECTED
2678      07E5      C2 F4 07      JNZ  RCA440      ;YES - SEE IF NEXT IS UNPROT
2679      07E8      2A C3 FF      LHLD CURADR      ;NO - RECALL CURRENT CHAR
2680      07EB      EB . .      XCHG              ;ADDRESS AND PUT INTO D,E
2681      07EC      . . .      RC4010 EQU $
2682      07EC      CD 90 0C      CALL NXTCHR      ;GET NEXT DISPLAY CHARACTER
2683      07EF      06 00 .      MVI  B,0         ;SET B FOR NOT PROTECTED
2684      07F1      C3 07 08      JMP  RCA460      ;EXIT CHARACTER FOUND
2685      07F4      . . .      ;*****
2686      07F4      . . .      ; PROTECT CHAR FOUND *
2687      07F4      . . .      ; SEE IF SUBSEQUENT UNPROTECT CHAR *
2688      07F4      . . .      ;*****
2689      07F4      . . .      RCA440 EQU $
2690      07F4      21 C2 C1      LXI  H,ENDPR*256+XMONLY ;IS NEXT CHARACTER
2691      07F7      CD 1E 20      CALL FNDCH      ;AN UNPROTECT OR XMIT ONLY
2692      07FA      . . .      ;*****
2693      07FA      . . .      ; ROM BREAK 1
2694      07FA      C3 02 08      JMP  ZBRK1C
2695      07FD      . . .      ORG  BEGIN+4000Q
2696      0800      . . .      ZBRK1 EQU $
2697      0800      54 . .      DB   VERSN      ;ROM PRESENT FLAGS
2698      0801      08 . .      DB   ZBRK1/256
2699      0802      . . .      ZBRK1C EQU $
2700      0802      . . .      ;*****
2701      0802      C2 EC 07      JNZ  RC4010      ;YES - RETURN UNPROTECTED
2702      0805      06 FF .      MVI  B,-1       ;NO - RETURN CHAR PROTECTED
2703      0807      . . .      RCA460 EQU $
2704      0807      21 C1 FF      LXI  H,CURCOL
2705      080A      4E . .      MOV  C,M        ;RECALL CURSOR COLUMN
2706      080B      . . .      ;
2707      080B      . . .      ; ZRETRN - RETURN WITH Z-FLAG TRUE
2708      080B      . . .      ;
2709      080B      . . .      ZRETRN EQU $
2710      080B      AF . .      XRA  A          ;SET ZERO FLAG
2711      080C      C9 . .      RET             ;RETURN
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 79
2713	080D	.	.	.	*****
2714	080D	.	.	.	; RCADDR - DETERMINE LOCATION OF ASCII CHARACTER *
2715	080D	.	.	.	; AT SPECIFIED ROW AND COLUMN OF DISPLAY LIST *
2716	080D	.	.	.	*****
2717	080D	.	.	.	;
2718	080D	.	.	.	; ENTRY: CURROW,CURCOL = DESIRED ROW/COLUMN
2719	080D	.	.	.	; LSTROW,LSTCOL = LAST ROW/COLUMN DONE
2720	080D	.	.	.	; CURADR = ADDRESS CORRESPONDING TO
2721	080D	.	.	.	; LSTROW, LSTCOL
2722	080D	.	.	.	; LSTLIN = ADDRESS OF LINE CORRESPONDING
2723	080D	.	.	.	; TO LSTROW
2724	080D	.	.	.	; NROWS = 0, BUILD NEW ROWS AS NEEDED
2725	080D	.	.	.	; # 0, DON'T BUILD NEW ROWS
2726	080D	.	.	.	; BLKFIL = 0, EXTEND LINE AS NEEDED
2727	080D	.	.	.	; # 0, DON'T EXTEND LINE
2728	080D	.	.	.	;
2729	080D	.	.	.	; EXIT : Z - CHARACTER FOUND
2730	080D	.	.	.	; A,B,C,L DESTROYED
2731	080D	.	.	.	; NZ - CHARACTER NOT FOUND
2732	080D	.	.	.	; M - ROWS NOT BUILT
2733	080D	.	.	.	; E = NUMBER OF ROWS NEEDED
2734	080D	.	.	.	; P - ROW LOCATED
2735	080D	.	.	.	; A = COLUMN NUMBER FOUND
2736	080D	.	.	.	; B = ROW NUMBER FOUND
2737	080D	.	.	.	; C = NUMBER OF CHARACTERS NEEDED
2738	080D	.	.	.	; U,E = ADDRESS OF LAST CHARACTER FOUND
2739	080D	.	.	.	; H = BASEH
2740	080D	.	.	.	;
2741	080D	.	.	.	; LSTROW,LSTCOL,LSTLIN,CURADR ARE UPDATED
2742	080D	.	.	.	; TO THE LAST CHARACTER FOUND.
2743	080D	.	.	.	;
2744	080D	.	.	.	RCADDR EQU \$
2745	080D	3A	C1	FF	LDA CURCOL ;GET DESIRED COLUMN NUMBER
2746	0810	.	.	.	RCAURO EQU \$
2747	0810	32	85	FF	STA TPCOL ;SAVE DESIRED COLUMN NUMBER
2748	0813	3A	C0	FF	LDA CURROW ;GET THE DESIRED ROW NUMBER
2749	0816	2A	C7	FF	LHLD LSTROW ;GET LAST ROW AND COLUMN DON
2750	0819	44	.	.	MOV B,H ;PUT LAST COLUMN IN B-REG
2751	081A	95	.	.	SUB L ;MOVED TO A NEW ROW?
2752	081B	2A	C9	FF	LHLD LSTLIN ;(GET LAST LINE DONE ADDR)
2753	081E	CA	63	08	JZ RCA240 ;YES - LOCATE COLUMN
2754	0821	.	.	.	*****
2755	0821	.	.	.	; ROW HAS CHANGED *
2756	0821	.	.	.	; LOCATE START OF NEW ROW *
2757	0821	.	.	.	*****
2758	0821	5F	.	.	MOV E,A ;SAVE COUNT
2759	0822	B7	.	.	ORA A ;SET FLAGS
2760	0823	F2	3A	08	JP RCA140 ;ROW IS AHEAD OF THIS ROW

ITFM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2762	0826	.	.	*****	80
2763	0826	.	.	; ROW IS BEFORE CURRENT ROW *	
2764	0826	.	.	; SEARCH BACK *	
2765	0826	.	.	*****	
2766	0826	.	.	RCA120 EQU \$	
2767	0826	23	.	INX H ;SET ADDRESS TO PREVIOUS	
2768	0827	23	.	INX H ;LINE POINTER	
2769	0828	CD	C6 1A	CALL CHAIN ;GET ADDRESS OF PREVIOUS ROW	
2770	0828	1C	.	INR E ;ROW FOUND?	
2771	082C	C2	26 08	JNZ RCA120 ;NO - CONTINUE BACKING UP	
2772	082F	C3	59 08	JMP RCA220 ;YES - SET NEW ROW	
2773	0832	.	.	*****	
2774	0832	.	.	; ROW IS AHEAD OF CURRENT ROW *	
2775	0832	.	.	; SEARCH AHEAD *	
2776	0832	.	.	*****	
2777	0832	.	.	RCA130 EQU \$	
2778	0832	CD	C6 1A	CALL CHAIN ;GET ADDRESS OF NEXT ROW	
2779	0835	23	.	INX H ;SET TO NEXT LINE PTR ADDRESS	
2780	0836	1D	.	DCR E ;ROW FOUND?	
2781	0837	CA	59 08	JZ RCA220 ;YES - LOCATE COLUMN	
2782	083A	.	.	RCA140 EQU \$;NO - CHECK FOR ANOTHER ROW	
2783	083A	7E	.	MOV A,M ;GET LSB OF NEXT ROW POINTER	
2784	083B	B7	.	ORA A ;DOES NEXT ROW EXIST?	
2785	083C	C2	32 08	JNZ RCA130 ;YES - CHECK FOR ROW FOUND	
2786	083F	.	.	*****	
2787	083F	.	.	; ROW NOT IN MEMORY *	
2788	083F	.	.	; CREATE NEW ROW *	
2789	083F	.	.	*****	
2790	083F	.	.	RCA200 EQU \$	
2791	083F	CD	CF 1A	CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE	
2792	0842	C2	0A 0C	JNZ NZEXIT ;YES - DO NOT BUILD ROWS	
2793	0845	21	9A FF	LXI H,NROWS ;NO - GET BUILD FLAG	
2794	0848	B6	.	ORA M ;INHIBIT ROW BUILD?	
2795	0849	C0	.	RNZ ;YES - RETURN (A = 377B)	
2796	084A	73	.	MOV M,E ;NO - STORE # OF ROWS NEEDED	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 81
=====
2798      084B      . . .      ;*****
2799      084B      . . .      ; GET NEW ROW AND LINK TO OLD *
2800      084B      . . .      ;*****
2801      084B      . . .      RCA210 EQU $
2802      084B      CD 77 06      CALL GTNWLN      ;ADD A LINE TO THE DISPLAY
2803      084E      C0 . .      RNZ              ;RETURN FAIL IF MEMORY LOCKE
2804      084F      21 9A FF      LXI H,NROWS     ;DECREMENT # OF ROWS NEEDED
2805      0852      35 . .      DCR M           ;ALL NEEDED ROWS ALLOCATED?
2806      0853      C2 4B 08      JNZ RCA210      ;NO - GET ANOTHER ROW
2807      0856      . . .      ;*****
2808      0856      . . .      ; ALL REQUIRED ROWS HAVE BEEN ADDED *
2809      0856      . . .      ;*****
2810      0856      2A A1 FF      LHL D LLINE     ;GET START ADDRESS OF ROW
2811      0859      . . .      RCA220 EQU $    ;UPDATE LOCATE COLUMN
2812      0859      CD A5 0B      CALL LSTLUP     ;SET "LSTLIN" TO NEW ROW
2813      085C      3A 85 FF      LDA TPCOL       ;RECALL COLUMN TO BE FOUND
2814      085F      4F . .      MOV C,A        ;PUT COLUMN NUMBER INTO C-RE
2815      0860      C3 6E 08      JMP RCA245      ;GO LOCATE THE COLUMN
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE	82
2817	0863	.	.	. ;*****		
2818	0863	.	.	. ; CURRENT ROW = DESIRED ROW *		
2819	0863	.	.	. ; CHECK COLUMN *		
2820	0863	.	.	. ;*****		
2821	0863	.	.	. RCA240 EQU \$		
2822	0863	3A	85	FF LDA TMPCOL ;GET THE DESIRED COLUMN		
2823	0866	4F	.	. MOV C,A ;PUT IT INTO THE C-REGISTER		
2824	0867	90	.	. SUB B ;COLUMN WANTED >= LAST DONE?		
2825	0868	F2	76	08 JP RCA250 ;YES - SCAN FORWARD		
2826	086B	.	.	. ;*****		
2827	086B	.	.	. ; DESIRED COLUMN LESS THAN CURRENT COLUMN *		
2828	086B	.	.	. ; START SEARCH AT BEGINNING OF ROW *		
2829	086B	.	.	. ;*****		
2830	086B	CD	A8	0B CALL LSTLU1 ;SET LINE START PARAMETERS		
2831	086E	.	.	. ; (PUTS H,L INTO D,E)		
2832	086E	.	.	. RCA245 EQU \$		
2833	086E	3E	01	. MVI A,IGNTRM ;SET FUNCTION FLAG TO IGNORE		
2834	0870	32	6D	FF STA TRMFCT ;NON-DISPLAYING TERMINATOR		
2835	0873	C3	80	08 JMP RCA255 ;GO LOCATE COLUMN		
2836	0876	.	.	. ;*****		
2837	0876	.	.	. ; DESIRED COLUMN AT OR PAST CURRENT COLUMN *		
2838	0876	.	.	. ; START SEARCH AT CURRENT COLUMN *		
2839	0876	.	.	. ;*****		
2840	0876	.	.	. RCA250 EQU \$		
2841	0876	4F	.	. MOV C,A ;SAVE # OF COLUMNS TO ADVANC		
2842	0877	2A	C3	FF LHLD CURADR ;GET ADDR OF LAST CHAR DONE		
2843	087A	EB	.	. XCHG		
2844	087B	04	.	. INR B ;DOES LSTCOL = -1?		
2845	087C	C2	81	08 JNZ RCA260 ;NO		
2846	087F	0D	.	. DCR C ;DECREMENT COLUMN COUNT		
2847	0880	.	.	. RCA255 EQU \$		
2848	0880	1B	.	. DCX D ;SET TO NEXT DISPLAY BYTE		

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2850	0881	.	.	*****	83
2851	0881	.	.	; ROW HAS BEEN FOUND *	
2852	0881	.	.	; SEARCH FOR DESIRED COLUMN *	
2853	0881	.	.	*****	
2854	0881	.	.	RCA260 EQU \$	
2855	0881	21	FE FF	LXI H,DISPST ;SET FOR NO CHARACTER MATCH	
2856	0884	CD	4D 20	CALL FNDCHR ;DOES CHARACTER EXIST?	
2857	0887	3E	00 .	MVI A,DELTRM ;SET FUNCTION FLAG TO DELETE	
2858	0889	32	6D FF	STA TRMFACT ;NON-DISPLAYING TERMINATOR	
2859	088C	CC	69 0A	CZ EOLMVO ;NO - TRY TO MOVE EOL	
2860	088F	EB	.	XCHG ;SET NEW CURRENT CHAR ADDRESS	
2861	0890	22	C3 FF	SHLD CURADR	
2862	0893	EB	.	XCHG	
2863	0894	21	C0 FF	LXI H,CURROW	
2864	0897	46	.	MOV B,M ;GET DESIRED ROW AND COLUMN	
2865	0898	3A	85 FF	LDA TMPCOL	
2866	089B	0D	.	DCR C ;CONVERT TO COLUMN FOUND	
2867	089C	FA	A0 08	JM RCA270	
2868	089F	91	.	SUB C	
2869	08A0	.	.	RCA270 EQU \$	
2870	08A0	68	.	MOV L,B ;UPDATE LAST ROW AND COLUMN	
2871	08A1	67	.	MOV H,A ;DONE	
2872	08A2	22	C7 FF	SHLD LSTROW	
2873	08A5	26	FF .	MVI H,BASEH ;SET H TO DATA PAGE	
2874	08A7	0C	.	INR C ;RESTORE ZERO FLAG	
2875	08A8	C9	.	RET ;RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2877	08A9	.	.	;*****	84
2878	08A9	.	.	; TIMER INTERRUPT PROCESSING *	
2879	08A9	.	.	;*****	
2880	08A9	.	.	;	
2881	08A9	.	.	; ENTRY: "PSW" AND B,C PUSHED	
2882	08A9	.	.	; A = INTERRUPT CODE	
2883	08A9	.	.	;	
2884	08A9	.	.	TMINTR EQU \$	
2885	08A9	CD	65 91	CALL INTVEC ;TRY ALTERNATE INTERRUPT	
2886	08AC	3A	F5 FF	LDA PRCCTL ;GET PROCESSOR STATE	
2887	08AF	D5	.	PUSH D ;SAVE REMAINING REGISTERS	
2888	08B0	E5	.	PUSH H	
2889	08B1	E6	FD .	ANI 377Q-TMIEN	
2890	08B3	D3	70 .	OUT PROCSR ;ACKNOWLEDGE TIMER INTERRUPT	
2891	08B5	F6	02 .	ORI TMIEN	
2892	08B7	D3	70 .	OUT PROCSR ;RE-ENABLE THE TIMER	
2893	08B9	21	D0 FF	LXI H,RSTTMR ;DECREMENT SOFT RESET DELAY	
2894	08BC	7E	.	MOV A,M ;TIMER	
2895	08BD	3D	.	DCR A ;COUNTING DOWN?	
2896	08BE	FA	C7 08	JM TMI010 ;NO - DON'T UPDATE TIMER	
2897	08C1	77	.	MOV M,A ;YES - STORE NEW VALUE	
2898	08C2	3E	06 .	MVI A,ENDTST ;(SET FOR RESET LED'S)	
2899	08C4	CC	08 48	CZ ZKBCTL ;RESET LED'S IF TIME DONE	
2900	08C7	.	.	TMI010 EQU \$	
2901	08C7	2E	50 .	MVI L,TPSTAL-BASE ;DECREMENT TAPE STALLED	
2902	08C9	7E	.	MOV A,M ;COUNTER	
2903	08CA	3D	.	DCR A ;STALL LIMIT REACHED?	
2904	08CB	FA	CF 08	JM TMI020 ;YES - DON'T UPDATE COUNTER	
2905	08CE	77	.	MOV M,A ;NO - STORE NEW VALUE	
2906	08CF	.	.	TMI020 EQU \$	
2907	08CF	2E	52 .	MVI L,CTBLTM-BASE ;DECREMENT BLINK TIMER	
2908	08D1	35	.	DCR M ;TIME OUT?	
2909	08D2	C2	E0 08	JNZ TMI100 ;NO - EXIT	
2910	08D5	36	20 .	MVI M,CTBDLY ;YES - RESET TIMER	
2911	08D7	23	.	INX H	
2912	08D8	7E	.	MOV A,M ;GET CTU BLINK MASK	
2913	08D9	2E	55 .	MVI L,CMND-BASE	
2914	08DB	AE	.	XRA M ;TOGGLE BLINKING LIGHTS	
2915	08DC	77	.	MOV M,A ;UPDATE LIGHT STATE	
2916	08DD	32	00 88	STA IOCTCO ;SET CTU LIGHTS	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2918	08E0	.	.	;*****	85
2919	08E0	.	.	; PERFORM KEYBOARD AND DATA COMM MONITOR *	
2920	08E0	.	.	; ROUTINES *	
2921	08E0	.	.	;*****	
2922	08E0	.	.	TMI100 EQU \$	
2923	08E0	21	F6	FF LXI H,INTFLG ;GET INTERRUPT FLAG	
2924	08E3	3E	04	. MVI A,TMRINT+1 ;TIMER INTERRUPT ALREADY	
2925	08E5	BE	.	. CMP M ;IN PROGRESS?	
2926	08E6	CA	FC	08 JZ TMI110 ;YES - DON'T DO MONITOR CALL	
2927	08E9	77	.	. MOV M,A ;NO - SET IN-PROGRESS FLAG	
2928	08EA	3A	7F	FE LDA DEVFLG ;GET DEVICE FLAGS	
2929	08ED	87	.	. ADD A ;ALTERNATE I/O INSTALLED?	
2930	08EE	FC	0B	92 CM ZMONAL ;YES - MONITOR ALT DEVICE	
2931	08F1	CD	0B	48 CALL ZKBMON	
2932	08F4	F3	.	. DI ;*****	
2933	08F5	CD	0E	50 CALL ZDCMON ;* KEYBOARD MONITOR ROUTINE	
2934	08F8	.	.	. ; * RE-ENABLES INTERRUPTS *	
2935	08F8	.	.	. ;*****	
2936	08F8	21	F6	FF LXI H,INTFLG ;SET INTERRUPT CODE TO	
2937	08FB	35	.	. DCR M ;INDICATE TIMER INTERRUPT	
2938	08FC	.	.	. TMI110 EQU \$	
2939	08FC	E1	.	. POP H ;RESTORE CONTENTS OF	
2940	08FD	D1	.	. POP D ;ALL REGISTERS AND	
2941	08FE	C1	.	. POP B ;ALL CONDITION FLAGS	
2942	08FF	F1	.	. POP PSW	
2943	0900	FB	.	. EI ;RE-ENABLE INTERRUPTS	
2944	0901	C9	.	. RET ;RETURN TO NORMAL PROCESSING	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	86
2946	0902	. . .	;*****		
2947	0902	. . .	; BINUCT - CONVERT BINARY TO OCTAL ASCII *		
2948	0902	. . .	;*****		
2949	0902	. . .	;		
2950	0902	. . .	; ENTRY: A = DIGIT TO BE CONVERTED		
2951	0902	. . .	; H,L = ADDRESS OF OUTPUT BUFFER'S		
2952	0902	. . .	; HIGH ORDER BYTE		
2953	0902	. . .	;		
2954	0902	. . .	; EXIT : H,L = H,L(ENTRY)+4		
2955	0902	. . .	; A-C DESTROYED		
2956	0902	. . .	;		
2957	0902	. . .	; FIRST BYTE IS SET TO BLANK. THE NEXT THREE		
2958	0902	. . .	; BYTES CONTAIN THE ASCII OCTAL EQUIVALENT OF		
2959	0902	. . .	; THE INPUT VALUE. THE FIFTH BYTE IS SET TO		
2960	0902	. . .	; ZERO (NULL).		
2961	0902	. . .	;		
2962	0902	. . .	BINUCT EQU \$		
2963	0902	36 20 .	MVI M,ABLNK ;SET FIRST BYTE TO BLANK		
2964	0904	23 . .	INX H		
2965	0905	06 03 .	MVI B,3 ;SET B TO NUMBER OF DIGITS		
2966	0907	07 . .	RLC ;ROTATE DOWN TWO HIGH ORDER		
2967	0908	07 . .	RLC ;BITS		
2968	0909	4F . .	MOV C,A ;SAVE VALUE IN C-REGISTER		
2969	090A	E6 03 .	ANI 30 ;MASK OUT TWO HIGH ORDER BIT		
2970	090C	. . .	BNO010 EQU \$		
2971	090C	E6 07 .	ANI 70 ;MASK OUT NEXT THREE BITS		
2972	090E	F6 30 .	ORI ZERO ;ADD IN ASCII ADJUSTMENT		
2973	0910	77 . .	MOV M,A ;STORE ASCII CHARACTER		
2974	0911	23 . .	INX H ;INCREMENT TO NEXT BYTE		
2975	0912	79 . .	MOV A,C ;RECALL INPUT		
2976	0913	07 . .	RLC ;ROTATE TO NEXT THREE BITS		
2977	0914	07 . .	RLC		
2978	0915	07 . .	RLC		
2979	0916	4F . .	MOV C,A ;SAVE VALUE		
2980	0917	05 . .	DCR B ;ALL BITS DONE?		
2981	0918	C2 0C 09	JNZ BNO010 ;NO - SET NEXT BYTE		
2982	0918	70 . .	MOV M,B ;YES - STORE NULL IN BUFFER		
2983	091C	C9 . .	RET ;RETURN		

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2985	0910	.	.	*****	87
2986	0910	.	.	; BN2DE0 - CONVERT SINGLE BYTE TO *	
2987	0910	.	.	; ASCII DECIMAL *	
2988	0910	.	.	*****	
2989	0910	.	.	;	
2990	0910	.	.	; ENTRY: A = BYTE TO BE CONVERTED	
2991	0910	.	.	; H,L = ADDRESS OF OUTPUT BUFFER'S	
2992	0910	.	.	; HIGH ORDER ADDRESS	
2993	0910	.	.	;	
2994	0910	.	.	; EXIT : NZ	
2995	0910	.	.	; H,L = H,L(ENTRY)+3	
2996	0910	.	.	; A-E DESTROYED	
2997	0910	.	.	;	
2998	0910	.	.	BN2DE0 EQU \$	
2999	0910	22	96	SHLD LNKS AV ;SAVE BUFFER ADDRESS	
3000	0920	21	6A	LXI H,B2D200 ;SET OUTPUT ROUTINE TO BUFFE	
3001	0923	.	.	*****	
3002	0923	.	.	BN2DA EQU \$	
3003	0923	.	.	*****	
3004	0923	.	.	BN2DE1 EQU \$;STORE ROUTINE	
3005	0923	22	CE	SHLD CNTFA D ;SET OUTPUT ROUTINE ADDRESS	
3006	0926	.	.	BN2DE2 EQU \$;ENTRY FOR "ASCOUT"	
3007	0926	5F	.	MOV E,A ;CHANGE INPUT INTO DOUBLE	
3008	0927	16	00	MVI D,0 ;BYTE VALUE	
3009	0929	0E	01	MVI C,1 ;SET ZERO SUPPRESS FLAG	
3010	092B	C3	45	JMP B2D050 ;GO TO CONVERT ROUTINE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  88
=====
3012     092E      . . .      ;*****
3013     092E      . . .      ; BN2DEC - CONVERT DOUBLE WORD BINARY TO DECIMAL *
3014     092E      . . .      ;*****
3015     092E      . . .      ;
3016     092E      . . .      ; ENTRY:  D,E = BINARY VALUE
3017     092E      . . .      ;           H,L = ADDRESS OF HIGH ORDER BYTE IN
3018     092E      . . .      ;           BUFFER
3019     092E      . . .      ;
3020     092E      . . .      ; EXIT :  H,L = H,L(ENTRY)+5
3021     092E      . . .      ;           A-E DESTROYED
3022     092E      . . .      ;           LNKSAV DESTROYED
3023     092E      . . .      ;
3024     092E      . . .      ; THE FIRST FIVE BYTES OF THE BUFFER CONTAIN THE
3025     092E      . . .      ; ASCII DECIMAL VALUE. THE SIXTH BYTE IS SET TO
3026     092E      . . .      ; ZERO (NULL). LEADING ZEROES ARE BLANKED
3027     092E      . . .      ;
3028     092E      . . .      BN2DEC EQU $
3029     092E      22 96 FF      SHLD LNKSAV      ;SAVE BUFFER ADDRESS
3030     0931      21 6A 09      LXI H,B2D200    ;SET OUTPUT ROUTINE TO BUFFE
3031     0934      . . .      ;*****
3032     0934      . . .      B2DDE EQU $
3033     0934      . . .      ;*****
3034     0934      22 CE FF      SHLD CNTFAD     ;STORE ROUTINE
3035     0937      0E 01 .      MVI C,1         ;SET ZERO SUPPRESS FLAG
3036     0939      21 F0 D8      LXI H,-10000
3037     093C      CD 58 09      CALL B2D100     ;EXTRACT 10,000'S VALUE
3038     093F      21 18 FC      LXI H,-1000
3039     0942      CD 58 09      CALL B2D100     ;EXTRACT 1,000'S VALUE
3040     0945      . . .      B2D050 EQU $
3041     0945      21 9C FF      LXI H,-100
3042     0948      CD 58 09      CALL B2D100     ;EXTRACT 100'S VALUE
3043     094B      21 F6 FF      LXI H,-10
3044     094E      CD 58 09      CALL B2D100     ;EXTRACT 10'S VALUE
3045     0951      7B . .      MOV A,E         ;CONVERT UNITS DIGIT TO
3046     0952      F6 30 .      ORI ZERO        ;ASCII AND STORE IN
3047     0954      0D . .      DCR C           ;SET C TO FORCE ZERO STORE
3048     0955      C3 CD FF      JMP ECONTF      ;GO TO OUTPUT ROUTINE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
3050	0958	.	.	;*****	89
3051	0958	.	.	; B2D100 - EXTRACT RADIX VALUE *	
3052	0958	.	.	;*****	
3053	0958	.	.	;	
3054	0958	.	.	; ENTRY: C = 0, SUPPRESS ZERO	
3055	0958	.	.	; < 0, DON'T SUPPRESS ZEROES	
3056	0958	.	.	; D,E = VALUE TO BE CONVERTED	
3057	0958	.	.	; H,L = -RADIX	
3058	0958	.	.	; LNKSAV = CURRENT BUFFER ADDRESS	
3059	0958	.	.	;	
3060	0958	.	.	; EXIT : C < 0, CHARACTER STORED	
3061	0958	.	.	; = 0, ZERO SUPPRESSED	
3062	0958	.	.	; (LNKSAV) = (LNKSAV)+1	
3063	0958	.	.	; A-C, H,L DESTROYED	
3064	0958	.	.	;	
3065	0958	.	.	B2D100 EQU \$	
3066	0958	06	2F	MVI B,ZERO-1 ;SET INITIAL ASCII VALUE	
3067	095A	EB	.	XCHG ;EXCHANGE RADIX AND INPUT	
3068	095B	.	.	B2D110 EQU \$	
3069	095B	04	.	INR B ;INCREMENT ASCII VALUE	
3070	095C	19	.	DAD D ;SUBTRACT RADIX	
3071	095D	DA	5B 09	JC B2D110 ;CONTINUE IF INPUT>RADIX	
3072	0960	7D	.	MOV A,L ;ADD BACK RADIX TO EXTRACT	
3073	0961	93	.	SUB E ;REMAINDER	
3074	0962	5F	.	MOV E,A ;SAVE REMAINDER IN D,E	
3075	0963	7C	.	MOV A,H	
3076	0964	9A	.	SBB D	
3077	0965	57	.	MOV D,A	
3078	0966	78	.	MOV A,B ;GET CONVERTED VALUE	
3079	0967	C3	CD FF	JMP ECONTF ;GO TO OUTPUT ROUTINE	
3080	096A	.	.	;*****	
3081	096A	.	.	; B2D200 - STORE DECIMAL VALUE FOR INTERNAL USE *	
3082	096A	.	.	;*****	
3083	096A	.	.	;	
3084	096A	.	.	; ENTRY: A = CONVERTED VALUE	
3085	096A	.	.	;	
3086	096A	.	.	B2D200 EQU \$	
3087	096A	FE	30	CPI ZERO ;CONVERTED VALUE = ZERO?	
3088	096C	C2	75 09	JNZ B2D210 ;NO - STORE THE DIGIT	
3089	096F	0D	.	DCR C ;NON-ZERO CHAR ALREADY DONE?	
3090	0970	FA	76 09	JM B2D220 ;YES - STORE THE DIGIT	
3091	0973	0C	.	INR C ;NO - RESTORE ZERO FLAG	
3092	0974	C9	.	RET ;AND EXIT	
3093	0975	.	.	B2D210 EQU \$	
3094	0975	0D	.	DCR C ;CLEAR ZERO SUPPRESS FLAG	
3095	0976	.	.	B2D220 EQU \$	
3096	0976	2A	96 FF	LHLD LNKSAV ;GET BUFFER POINTER	
3097	0979	77	.	MOV M,A ;STORE CONVERTED VALUE	
3098	097A	23	.	INX H ;INCREMENT BUFFER POINTER	
3099	097B	36	00	MVI M,0 ;SET NEXT BYTE TO NULL	

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

```
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 90
3100	097D	22 96 FF	SHLD LNKSAV ;STORE NEW POINTER VALUE	
3101	0980	C9 . .	RET ;RETURN	

```
=====
```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
3103	0981	. . .	;*****	91
3104	0981	. . .	; CALCULATE CHECKSUM	
3105	0981	. . .	;	
3106	0981	. . .	; ENTRY	
3107	0981	. . .	; (H,L) = ADDRESS OF AREA	
3108	0981	. . .	; TO BE CHECKSUMED	
3109	0981	. . .	;	
3110	0981	. . .	; D = NO. BYTES IN AREA/256	
3111	0981	. . .	; WE ASSUME THE AREA BEGINS ON A	
3112	0981	. . .	; 256 BYTE BOUNDARY, I.E., L=0.	
3113	0981	. . .	; CALL CHKSUM	
3114	0981	. . .	; EXIT	
3115	0981	. . .	; A = CHECKSUM	
3116	0981	. . .	; ALL OTHER REGS. UNCHANGED	
3117	0981	. . .	; FLAGS DESTROYED	
3118	0981	. . .	;*****	
3119	0981	. . .	CHKSUM EQU \$	
3120	0981	D5 . .	PUSH D ;SAVE REGISTER D-H	
3121	0982	E5 . .	PUSH H	
3122	0983	AF . .	XRA A ;ZERO SUM	
3123	0984	. . .	CSU100 EQU \$	
3124	0984	86 . .	ADD M ;ADD BYTE	
3125	0985	CE 00 .	ACI 0 ;ADD CARRY	
3126	0987	2C . .	INR L ;BUMP ADDRESS POINTER	
3127	0988	C2 84 09	JNZ CSU100 ;ADD NEXT BYTE	
3128	0988	. . .	;	
3129	0988	24 . .	INR H ;FINISHED A 256 BYTE BLOCK	
3130	098C	15 . .	DCR D	
3131	098D	C2 84 09	JNZ CSU100 ;DO NEXT 256 BYTES	
3132	0990	. . .	;	
3133	0990	03 . .	INX B ;INCREMENT TO NEXT STORE ADD	
3134	0991	57 . .	MOV D,A ;SAVE CHECKSUM IN D-REGISTER	
3135	0992	E1 . .	POP H ;RECALL STARTING ADDRESS	
3136	0993	7C . .	MOV A,H	
3137	0994	FE F0 .	CPI 170000Q/256 ;LAST RAM BLOCK?	
3138	0996	C2 9A 09	JNZ CSU110 ;NO - EXIT	
3139	0999	4D . .	MOV C,L ;YES - SET B,C TO FIRST	
3140	099A	. . .	; CHECKSUM STORE ADDRESS	
3141	099A	. . .	CSU110 EQU \$	
3142	099A	7A . .	MOV A,D ;PUT CHECKSUM BACK INTO A-RE	
3143	099B	D1 . .	POP D ;RESTORE D,E	
3144	099C	C9 . .	RET ;RETURN	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  92
=====
3146      099D      . . .      ;*****
3147      099D      . . .      ; CLEAR - RESET TERMINAL BY ESCAPE SEQUENCE *
3148      099D      . . .      ;*****
3149      099D      . . .      CLEAR EQU $
3150      099D      CD C0 16    CALL IOBSYC ;WAIT UNTIL TAPES NOT BUSY
3151      09A0      3E 04 .     MVI A,FRCRST ;SET FLAG TO FORCE FULL
3152      09A2      CD 44 15    CALL STCMFL ;TERMINAL RESET
3153      09A5      3E 80 .     MVI A,CRTOFF ;TURN OFF THE DISPLAY
3154      09A7      32 20 87    STA IOCRRW
3155      09AA      C7 . .      RST RESET ;GO DO TERMINAL RESET
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 93
3157	09AB	. . .	;*****	
3158	09AB	. . .	; DISPL1 - ADD ENOUGH BLOCKS TO REACH DESIRED *	
3159	09AB	. . .	; COLUMN *	
3160	09AB	. . .	;*****	
3161	09AB	. . .	;	
3162	09AB	. . .	; ENTRY: C = NUMBER OF CHARACTERS NEEDED - 1	
3163	09AB	. . .	; D,E = LOCATION OF EOL IN LINE	
3164	09AB	. . .	;	
3165	09AB	. . .	; EXIT : A = 0, NOT ENOUGH BLOCKS (MEMORY LOCK)	
3166	09AB	. . .	; B-L DESTROYED	
3167	09AB	. . .	; A # 0, MEMORY ALLOCATED	
3168	09AB	. . .	; D,E = FIRST CHAR ADDR IN NEW BLOCKS	
3169	09AB	. . .	; B,C,H,L DESTROYED	
3170	09AB	. . .	;	
3171	09AB	. . .	; IF ONLY ONE CHARACTER IS TO BE ADDED, THE	
3172	09AB	. . .	; CHARACTER IS ADDED TO THE LINE. OTHERWISE, ALL	
3173	09AB	. . .	; REQUIRED BLOCKS ARE ADDED TO THE LINE AND THE	
3174	09AB	. . .	; LINE IS FILLED WITH BLANKS UP TO THE DESIRED	
3175	09AB	. . .	; CHARACTER ONLY.	
3176	09AB	. . .	;	
3177	09AB	. . .	DISPL1 EQU \$	
3178	09AB	0C . .	INR C ;MOVE EOL IF NECESSARY	
3179	09AC	CD 71 0A	CALL EOLMOV	
3180	09AF	0D . .	DCR C	
3181	09B0	FA 4A 0A	JM DIS220 ;CHARACTER POSITION FOUND	
3182	09B3	21 9B FF	LXI H,NCHAR ;SAVE NUMBER OF CHARACTERS	
3183	09B6	71 . .	MOV M,C ;TO BE ADDED - 1	
3184	09B7	. . .	DISPL2 EQU \$	
3185	09B7	EB . .	XCHG	
3186	09B8	22 94 FF	SHLD EOLADR ;SAVE EOL ADDRESS	
3187	09BB	0D . .	DCR C ;SINGLE CHARACTER ADDED?	
3188	09BC	FA 5A 0A	JM DIS400 ;YES - DO FAST EXTEND	
3189	09BF	3E 20 .	MVI A,ABLNK ;NO - GET A DISPLAY BLOCK	
3190	09C1	CD 2D 06	CALL GTBLK ;FILLED WITH BLANKS	
3191	09C4	C8 . .	RZ ;RETURN IF MEMORY LOCKED	
3192	09C5	EB . .	XCHG ;PUT BLOCK ADDRESS IN D,E	
3193	09C6	F6 0F .	ORI BLKSM ;COMPUTE HIGH ADDR OF BLOCK	
3194	09C8	4F . .	MOV C,A ;SAVE ADDRESS OF FIRST NEW	
3195	09C9	C5 . .	PUSH B ;BLOCK ADDED	
3196	09CA	3A 9B FF	LDA NCHAR ;GET # OF CHARS TO BE ADDED	
3197	09CD	06 00 .	MVI B,0 ;INITIALIZE COUNT	
3198	09CF	. . .	DIS120 EQU \$	
3199	09CF	04 . .	INR B ;INCREMENT COUNT	
3200	09D0	D6 0E .	SUI BLKSM-2 ;SUB. NO. OF CHARS IN BLOCK	
3201	09D2	F2 CF 09	JP DIS120 ;JUMP IF MORE BLOCKS NEEDED	
3202	09D5	32 84 FF	STA COUNT ;SAVE LAST CHAR BLOCK POS	
3203	09D8	05 . .	DCR B ;SINGLE BLOCK?	
3204	09D9	CA FA 09	JZ DIS160 ;YES	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 94
=====
3206      09DC      . . .      ;*****
3207      09DC      . . .      ; MULTIPLE BLOCKS REQUIRED *
3208      09DC      . . .      ;*****
3209      09DC      21 99 FF      LXI H,NBLKS      ;SAVE BLOCK COUNT
3210      09DF      70 . .      MOV M,B
3211      09E0      . . .      ;*****
3212      09E0      . . .      ; GET SUBSEQUENT BLOCKS *
3213      09E0      . . .      ;*****
3214      09E0      D5 . .      PUSH D           ;SAVE ADDRESS OF LAST BLOCK
3215      09E1      . . .      DIS140 EQU $
3216      09E1      3E 20 .      MVI A,ABLNK     ;GET A DISPLAY BLOCK FILLED
3217      09E3      CD 2D 06      CALL GTRLK      ;WITH BLANKS
3218      09E6      EB . .      XCHG           ;PUT BLOCK ADDRESS IN D,E
3219      09E7      E1 . .      POP H          ;RECALL ADDRESS OF LAST BLOC
3220      09E8      CA 54 0A      JZ DIS240      ;EXIT IF MEMORY LOCKED
3221      09EB      D5 . .      PUSH D         ;SAVE NEW LINE ADDRESS
3222      09EC      2B . .      DCX H         ;LINK NEW BLOCK TO PREVIOUS
3223      09ED      F6 0F .      ORI BLKSM
3224      09EF      70 . .      MOV M,B       ;MSB'S
3225      09F0      2B . .      DCX H
3226      09F1      77 . .      MOV M,A       ;STORE LSB
3227      09F2      21 99 FF      LXI H,NBLKS
3228      09F5      35 . .      DCR M         ;ALL BLOCKS ALLOCATED?
3229      09F6      C2 E1 09      JNZ DIS140    ;NO - GET ANOTHER BLOCK
3230      09F9      F1 . .      POP PSW       ;YES - POP THE STACK
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 95
3232	09FA	.	.	;*****	
3233	09FA	.	.	; ALL BLOCKS HAVE BEEN ADDED *	
3234	09FA	.	.	;*****	
3235	09FA	.	.	DIS160 EQU \$	
3236	09FA	3A	84 FF	LDA COUNT ;COMPUTE NUMBER OF BYTES	
3237	09FD	2F	.	CMA ;TO FILL	
3238	09FE	3C	.	INR A	
3239	09FF	4F	.	MOV C,A ;SAVE IN C	
3240	0A00	83	.	ADD E ;GET FIRST FILL ADDR	
3241	0A01	3D	.	DCR A ;SET FIRST LSB FILL ADDRESS	
3242	0A02	6F	.	MOV L,A ;PUT LSB INTO L	
3243	0A03	62	.	MOV H,D ;GET MSB FROM D	
3244	0A04	06 CC	.	MVI B,EOL ;SET "EOL" CHARACTER	
3245	0A06	3A C1 FF	.	LDA CURCOL ;GET CURRENT COLUMN	
3246	0A09	FE 4F	.	CPI MAXCOL ;CHAR ADDED TO LAST COLUMN?	
3247	0A0B	C2 15 0A	.	JNZ DIS170 ;NO - SET "EOL" CHARACTER	
3248	0A0E	3A 89 FF	.	LDA DCHAR ;YES - GET CHARACTER STORED	
3249	0A11	B7	.	ORA A ;IS IT ASCII?	
3250	0A12	F2 16 0A	.	JP DIS175 ;YES - DON'T ADD "EOL"	
3251	0A15	.	.	; NO - SET "EOL" CHARACTER	
3252	0A15	.	.	;*****	
3253	0A15	.	.	; FILL UNUSED PART OF BLOCK WITH "FILL" CODES *	
3254	0A15	.	.	;*****	
3255	0A15	.	.	DIS170 EQU \$	
3256	0A15	70	.	MOV M,B ;STORE FILL/EOL CHARACTER	
3257	0A16	.	.	DIS175 EQU \$	
3258	0A16	2B	.	DCX H ;GO TO NEXT BYTE	
3259	0A17	0D	.	DCR C ;ALL UNUSED BYTES FILLED?	
3260	0A18	06 C3	.	MVI B,FILL ;(SET "FILL" CODE)	
3261	0A1A	C2 15 0A	.	JNZ DIS170 ;NO - SET NEXT BYTE	
3262	0A1D	.	.	;*****	
3263	0A1D	.	.	; WRITE LINK TO NEXT LINE *	
3264	0A1D	.	.	;*****	
3265	0A1D	.	.	DIS180 EQU \$	
3266	0A1D	2A C9 FF	.	LHLD LSTLIN ;GET ADDR CURRENT LINE	
3267	0A20	EB	.	XCHG	
3268	0A21	2B	.	DCX H ;STORE AS NEXT BLOCK POINTER	
3269	0A22	72	.	MOV M,D	
3270	0A23	2B	.	DCX H	
3271	0A24	13	.	INX D ;POINT TO NEXT LINE POINTER	
3272	0A25	73	.	MOV M,E	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  96
=====
3274      0A26      .      .      .      ;*****
3275      0A26      .      .      .      ; LINK NEW BLOCK(S) TO OLD *
3276      0A26      .      .      .      ;*****
3277      0A26      D1      .      .      POP D      ;RECALL FIRST NEW BLOCK ADDR
3278      0A27      3A      9B      FF      LDA NCHAR  ;GET # OF CHARS ADDED - 1
3279      0A2A      B7      .      .      ORA A      ;DOES NEW CHAR REPLACE EOL?
3280      0A2B      3A      89      FF      LDA DCHAR  ;(DEFAULT TO ADD 1 CHAR)
3281      0A2E      CA      33      0A      JZ DIS210  ;YES - OVERWRITE EOL
3282      0A31      3E      20      .      MVI A,ABLNK ;NO - STORE BLANK OVER EOL
3283      0A33      .      .      .      DIS210 EQU $
3284      0A33      47      .      .      MOV B,A    ;SAVE CHARACTER TO BE STORED
3285      0A34      2A      94      FF      LHLD EOLADR ;RECALL EOL ADDRESS
3286      0A37      3A      C0      FF      LDA CURROW
3287      0A3A      F6      40      .      ORI MAYEOL ;SET FOR POSSIBLE EOL SKIP
3288      0A3C      F3      .      .      DI        ;DISABLE INTERRUPTS
3289      0A3D      .      .      .      ;!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!*
3290      0A3D      CD      0B      60      CALL ZANCHK ;TURN OFF DISPLAY DMA
3291      0A40      .      .      .      ;*****
3292      0A40      70      .      .      MOV M,B    ;OVERWRITE EOL
3293      0A41      2B      .      .      DCX H
3294      0A42      72      .      .      MOV M,D    ;CHANGE NEXT BLOCK LINK TO
3295      0A43      2B      .      .      DCX H      ;POINT TO NEW BLOCKS
3296      0A44      73      .      .      MOV M,E
3297      0A45      CD      D3      10      CALL DISLN1 ;TURN DISPLAY BACK ON
3298      0A48      B4      .      .      ORA H      ;SET Z-FALSE
3299      0A49      C9      .      .      RET        ;RETURN
3300      0A4A      .      .      .      ;*****
3301      0A4A      .      .      .      ; EOL MOVE SATISFIED REQUEST *
3302      0A4A      .      .      .      ; CHECK FOR SINGLE CHARACTER *
3303      0A4A      .      .      .      ;*****
3304      0A4A      .      .      .      DIS220 EQU $
3305      0A4A      3D      .      .      DCR A      ;SINGLE CHARACTER?
3306      0A4B      32      9B      FF      STA NCHAR  ;(SET NCHAR)
3307      0A4E      C0      .      .      RNZ        ;NO - RETURN
3308      0A4F      3A      89      FF      LDA DCHAR  ;YES - GET THE CHARACTER
3309      0A52      12      .      .      STAX D     ;STORE THE CHARACTER
3310      0A53      C9      .      .      RET        ;RETURN
3311      0A54      .      .      .      ;*****
3312      0A54      .      .      .      ; ALL BLOCKS NOT AVAILABLE *
3313      0A54      .      .      .      ; INITIALIZE END OF LINE *
3314      0A54      .      .      .      ;*****
3315      0A54      .      .      .      DIS240 EQU $
3316      0A54      36      CC      .      MVI M,EOL  ;STORE AN EOL
3317      0A56      E8      .      .      XCHG      ;PUT ADDRESS INTO D,E
3318      0A57      C3      1D      0A      JMP DIS180
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 97
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
3320	0A5A	.	.	;*****	
3321	0A5A	.	.	; SINGLE CHARACTER ADDITION *	
3322	0A5A	.	.	;*****	
3323	0A5A	.	.	DIS400 EQU \$	
3324	0A5A	CD	2B 06	CALL GTBLKF ;GET A DISPLAY BLOCK	
3325	0A5D	C8	.	RZ ;RETURN IF MEMORY LOCKED	
3326	0A5E	54	.	MOV D,H ;SAVE BLOCK ADDRESS IN D,E	
3327	0A5F	5D	.	MOV E,L	
3328	0A60	F6	0F .	ORI BLKSM ;PUT AN EOL AT THE FIRST	
3329	0A62	6F	.	MOV L,A ;DISPLAY CHARACTER	
3330	0A63	36	CC .	MVI M,EOL ;LOCATION IN THE BLOCK	
3331	0A65	E5	.	PUSH H ;SAVE ADDRESS OF BLOCK	
3332	0A66	C3	1D 0A	JMP DIS180 ;LINK BLOCK TO DISPLAY	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
3334	0A69	.	.	;	98
3335	0A69	.	.	; * * * * *	
3336	0A69	.	.	;	
3337	0A69	.	.	; EOLMOV - MOVE EOL IN A BLOCK	
3338	0A69	.	.	;	
3339	0A69	.	.	; ENTRY: C = NUMBER OF BYTES NEEDED	
3340	0A69	.	.	; D,E = ADDRESS OF EXISTING EOL	
3341	0A69	.	.	;	
3342	0A69	.	.	; EXIT : A = NUMBER OF CHARACTERS ADDED	
3343	0A69	.	.	; C = 0, CHARACTER FOUND	
3344	0A69	.	.	; D,E = ADDRESS OF CHARACTER	
3345	0A69	.	.	; C = NUMBER OF CHARACTERS NEEDED	
3346	0A69	.	.	; D,E = ADDRESS OF LAST BYTE IN BLK	
3347	0A69	.	.	; H = BASEH	
3348	0A69	.	.	; B,L DESTROYED	
3349	0A69	.	.	;	
3350	0A69	.	.	; EOLMVO - MOVE ONLY IF UNPROTECTED	
3351	0A69	.	.	;	
3352	0A69	.	.	EOLMVO EQU \$	
3353	0A69	3A	91	LDA BLKFIL ;GET BLOCK FILL INHIBIT FLAG	
3354	0A6C	3C	.	INR A ;BLOCK FILL INHIBITED OR	
3355	0A6D	C4	90	11 CNZ CKPROT ;CURSOR IN PROTECTED FIELD	
3356	0A70	C8	.	RZ ;YES - RETURN	
3357	0A71	.	.	;	
3358	0A71	.	.	EOLMOV EQU \$	
3359	0A71	7B	.	MOV A,E ;COMPUTE NUMBER OF BYTES	
3360	0A72	E6	0F	ANI BLKSM ;AVAILABLE IN BLOCK	
3361	0A74	D6	02	SUI 2 ;(DELETE BYTES FOR LINK)	
3362	0A76	C8	.	RZ ;RETURN IF NONE AVAILABLE	
3363	0A77	EB	.	XCHG ;PUT CURRENT ADDRESS IN H,L	
3364	0A78	B9	.	CMP C ;ENOUGH CHARACTERS?	
3365	0A79	47	.	MOV B,A ;(SET B TO FILL BLOCK)	
3366	0A7A	11	40	CC LXI D,EOL*256+MAYEOL ;(SET FOR PARTIAL	
3367	0A7D	.	.	; LINE EXTENSION)	
3368	0A7D	FA	92	0A JM ELM100 ;NO - BLANK REST OF BLOCK	
3369	0A80	41	.	MOV B,C ;YES - BLANK WHAT'S NEEDED	
3370	0A81	3A	C1	FF LDA CURCOL ;GET CURRENT COLUMN POSITION	
3371	0A84	FE	4F	. CPI MAXCOL ;ADDING TO LAST COLUMN?	
3372	0A86	C2	92	0A JNZ ELM100 ;NO - NEED EOL AT LINE END	
3373	0A89	3A	89	FF LDA DCHAR ;YES - GET NEW CHARACTER	
3374	0A8C	B7	.	ORA A ;IS IT ASCII?	
3375	0A8D	FA	92	0A JM ELM100 ;NO - NEED EOL AT LINE END	
3376	0A90	16	C3	. MVI D,FILL ;YES - DON'T NEED EOL	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
3378	0A92	.	.	;	99
3379	0A92	.	.	; FILL THE BLOCK	
3380	0A92	.	.	;	
3381	0A92	.	.	ELM100 EQU \$	
3382	0A92	79	.	MOV A,C ;COMPUTE NUMBER OF ADDITIONA	
3383	0A93	90	.	SUB B ;BYTES NEEDED	
3384	0A94	4F	.	MOV C,A ;SAVE IT IN C FOR RETURN	
3385	0A95	3A	C0	LDA CURROW ;SET CONTROL TO TURN OFF DMA	
3386	0A98	B3	.	ORA E	
3387	0A99	58	.	MOV E,B ;SAVE NUMBER OF BYTES ADDED	
3388	0A9A	F3	.	DI ;DISABLE INTERRUPTS	
3389	0A9B	.	.	*****	
3390	0A9B	CD	0B	CALL ZANCHK ;TURN OFF DMA	
3391	0A9E	.	.	*****	
3392	0A9E	.	.	;	
3393	0A9E	.	.	ELM110 EQU \$	
3394	0A9E	36	20	MVI M,ABLNK ;FILL BLOCK WITH BLANKS	
3395	0AA0	2B	.	DCX H ;MOVE TO NEXT BYTE	
3396	0AA1	05	.	DCR B ;FILL COMPLETED?	
3397	0AA2	C2	9E	JNZ ELM110 ;NO - DO NEXT BYTE	
3398	0AA5	72	.	MOV M,D ;YES - ADD EOL OR EOL FILL	
3399	0AA6	CD	D3	CALL DISLN1 ;TURN DISPLAY BACK ON	
3400	0AA9	AF	.	XRA A ;CLEAR A-REGISTER	
3401	0AAA	B1	.	ORA C ;ALL CHARACTERS DONE?	
3402	0AAB	C2	AF	JNZ ELM130 ;NO - RETURN ADDRESS OF EOL	
3403	0AAE	23	.	INX H ;YES - RETURN ADDR OF LAST C	
3404	0AAF	.	.	ELM130 EQU \$	
3405	0AAF	7B	.	MOV A,E ;PUT # OF CHARS DONE IN A-RE	
3406	0AB0	EB	.	XCHG ;PUT CHARACTER ADDRESS IN D,	
3407	0AB1	21	90	LXI H,EOLMV ;(SET H TO DATA PAGE)	
3408	0AB4	36	01	MVI M,1 ;SET EOLMV FLAG	
3409	0AB6	C9	.	RET ;RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 100
3411	0AB7	.	.	*****	
3412	0AB7	.	.	; LD - LINE DELETE *	
3413	0AB7	.	.	*****	
3414	0AB7	.	.	LINDEL EQU \$	
3415	0AB7	CD	CF 1A	CALL CHKFMS ;FORMAT MODE?	
3416	0ABA	CC	B1 07	CZ RCADR1 ;FIND LINE IF NOT	
3417	0ABD	C0	.	RNZ ;LINE NOT FOUND	
3418	0ABE	2A	C9 FF	LHLD LSTLIN ;GET ADDR OF LAST LINE DONE	
3419	0AC1	7E	.	MOV A,M ;GET PREVIOUS LINE'S LSB	
3420	0AC2	B7	.	ORA A ;ANY PREVIOUS LINES?	
3421	0AC3	CA	D4 0A	JZ LID050 ;NO - DO CLEAR LINE ONLY	
3422	0AC6	CD	DA 0A	CALL LINDLO ;YES - DELETE CURRENT LINE	
3423	0AC9	.	.	*****	
3424	0AC9	.	.	; UPDATE LSTLIN AND CURADR TO ADDRESS *	
3425	0AC9	.	.	; OF NEXT LINE *	
3426	0AC9	.	.	*****	
3427	0AC9	60	.	MOV H,B ;PUT NEW LINE ADDRESS INTO	
3428	0ACA	69	.	MOV L,C ;H,L	
3429	0ACB	CD	41 1A	CALL BACKTS ;UPDATE CURRENT LINE AND ADD	
3430	0ACE	CD	27 0B	CALL LININO ;GO UPDATE TOP LINE IF NEEDED	
3431	0AD1	C3	8E 07	JMP PUTLIN ;ADD LINE TO FREE LIST	
3432	0AD4	.	.	LID050 EQU \$	
3433	0AD4	CD	95 1D	CALL CLEARL ;CLEAR THE LINE	
3434	0AD7	C3	7C 23	JMP CURPRT ;SET CURSOR AT LEFT MARGIN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 101
3436	OADA	.	.	*****	
3437	OADA	.	.	; LINDLO - RMOVE LINE FROM LINKED LIST *	
3438	OADA	.	.	*****	
3439	OADA	.	.	;	
3440	OADA	.	.	; ENTRY: H,L = ADDRESS OF NEXT LINE FIELD	
3441	OADA	.	.	; (LSB) OF LINE TO BE DELETED	
3442	OADA	.	.	;	
3443	OADA	.	.	; EXIT : B,C = ADDRESS OF LSB PORTION OF	
3444	OADA	.	.	; NEXT LINE POINTER IN NEW LINE	
3445	OADA	.	.	; D,E = H,L(ENTRY)	
3446	OADA	.	.	; A,H,L DESTROYED	
3447	OADA	.	.	;	
3448	OADA	.	.	LINDLO EQU \$	
3449	OADA	5D	.	MOV E,L ;SAVE ADDRESS OF LINE TO BE	
3450	OADB	54	.	MOV D,H ;DELETED IN D,E	
3451	OADC	4E	.	MOV C,M ;GET ADDRESS OF NEXT LINE	
3452	OADD	23	.	INX H	
3453	OADE	46	.	MOV B,M	
3454	OADF	23	.	INX H ;GET ADDRESS OF PREVIOUS LIN	
3455	OAE0	7E	.	MOV A,M	
3456	OAE1	23	.	INX H	
3457	OAE2	66	.	MOV H,M	
3458	OAE3	87	.	ORA A ;DOES PREVIOUS LINE EXIST?	
3459	OAE4	C2 F0 0A	.	JNZ LID200 ;YES - LINK 2 LINES TOGETHER	
3460	OAE7	.	.	*****	
3461	OAE7	.	.	; FIRST LINE DELETED - UPDATE FLINE *	
3462	OAE7	.	.	*****	
3463	OAE7	60	.	MOV H,B ;MOVE NEW CURRENT LINE TO H,	
3464	OAE8	69	.	MOV L,C	
3465	OAE9	23	.	INX H ;SET ADDR TO NEXT LINE FIELD	
3466	OAEA	22 9F FF	.	SHLD FLINE	
3467	OAE0	C3 F6 0A	.	JMP LID300 ;SET NEW PREV LINE POINTER	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 102
=====
3469     0AF0      . . .      ;*****
3470     0AF0      . . .      ; UPDATE NEXT LINE FIELD IN PREVIOUS LINE *
3471     0AF0      . . .      ;*****
3472     0AF0      . . .      LID200 EQU $
3473     0AF0      6F . .      MOV L,A      ;PUT LSB INTO L-REGISTER
3474     0AF1      23 . .      INX H        ;SET TO MSB OF NEXT LINE FLD
3475     0AF2      CD CA 10    CALL DISLNK  ;SET NEW NEXT LINE LINK TO
3476     0AF5      . . .      ;
3477     0AF5      . . .      ;*****
3478     0AF5      . . .      ; SET PREVIOUS LINE FIELD IN NEXT LINE *
3479     0AF5      . . .      ;*****
3480     0AF5      7D . .      MOV A,L      ;SAVE PREV LINE ADDR'S LSB
3481     0AF6      . . .      LID300 EQU $
3482     0AF6      03 . .      INX B        ;INCREMENT TO NEXT LINE PTR
3483     0AF7      C5 . .      PUSH B       ;SAVE ADDRESS
3484     0AF8      03 . .      INX B        ;SET ADDRESS TO PREVIOUS
3485     0AF9      03 . .      INX B        ;LINE FIELD
3486     0AFA      02 . .      STAX B       ;STORE LSB VALUE
3487     0AFB      03 . .      INX B
3488     0AFC      7C . .      MOV A,H
3489     0AFD      02 . .      STAX B       ;STORE MSB VLAUE
3490     0AFE      C1 . .      POP B        ;RESTORE CONTENTS OF B,C
3491     0AFF      C9 . .      RET          ;RETURN
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 103
3493	0B00	.	.	*****	
3494	0B00	.	.	; LI - LINE INSERT *	
3495	0B00	.	.	*****	
3496	0B00	.	.	LININS EQU \$	
3497	0B00	CD	CF 1A	CALL CHKFMS ;FORMAT MODE?	
3498	0B03	CC	B1 07	CZ RCADR1 ;FIND LINE IF NOT	
3499	0B06	C0	.	RNZ ;RETURN IF LINE NOT FOUND	
3500	0B07	CD	2B 06	CALL GTBLKF ;GET BLOCK FOR NEW LINE	
3501	0B0A	C8	.	RZ ;RETURN IF NOT AVAILABLE	
3502	0B0B	.	.	*****	
3503	0B0B	.	.	; STORE LINK AT END OF NEW LINE *	
3504	0B0B	.	.	*****	
3505	0B0B	C6	0B .	ADI BLKSZ-5 ;GET ADDR OF NEXT LINE FIELD	
3506	0B0D	2D	.	DCR L	
3507	0B0E	74	.	MOV M,H ;STORE LINK MSB'S	
3508	0B0F	2D	.	DCR L	
3509	0B10	77	.	MOV M,A ;STORE LINK LSB'S	
3510	0B11	D6	02 .	SUI 2 ;STORE EOL IN NEW LINE	
3511	0B13	6F	.	MOV L,A	
3512	0B14	CD	79 0E	CALL STCHR1 ;SET FIRST DISPLAY CHARACTER	
3513	0B17	.	.	*****	
3514	0B17	.	.	; ADJUST LSTLIN AND CURADR PNTRS TO NEW LINE *	
3515	0B17	.	.	*****	
3516	0B17	22	C3 FF	SHLD CURADR ;SET CURADR TO 1ST CHAR	
3517	0B1A	23	.	INX H ;SET TO NEXT LINE POINTER	
3518	0B1B	7D	.	MOV A,L ;PUT LSB INTO A-REGISTER	
3519	0B1C	EB	.	XCHG	
3520	0B1D	2A	C9 FF	LHLD LSTLIN ;GET CURRENT LINE ADDRESS	
3521	0B20	EB	.	XCHG	
3522	0B21	22	C9 FF	SHLD LSTLIN ;SET NEW CURRENT LINE ADDRESS	
3523	0B24	CD	3C 0B	CALL LININ1 ;ADD LINE TO DISPLAY LIST	
3524	0B27	.	.	*****	
3525	0B27	.	.	; UPDATE TOPLIN IF ROW ZERO *	
3526	0B27	.	.	*****	
3527	0B27	.	.	LININ0 EQU \$	
3528	0B27	CD	A9 0B	CALL LSTLU2 ;SET INITIAL LINE STATE	
3529	0B2A	CD	7C 23	CALL CURPRT ;SET CURSOR TO LEFT MARGIN	
3530	0B2D	AF	.	XRA A ;SET LAST COLUMN DONE TO	
3531	0B2E	32	C8 FF	STA LSTCOL ;ZERO	
3532	0B31	21	C0 FF	LXI H,CURROW ;GET CURRENT ROW NUMBER	
3533	0B34	B6	.	DRA M ;DID TOP ROW CHANGE?	
3534	0B35	C0	.	RNZ ;NO - RETURN	
3535	0B36	C3	B0 10	JMP TOPUP1 ;YES - UPDATE TOP LINE VALUE	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 104
3537	0B39	. . .	;*****	
3538	0B39	. . .	; LININ1 - ADD LINE TO LINK LIST *	
3539	0B39	. . .	; ENTRY D,E=NEXT PAGE FIELD ADDR IN LINE *	
3540	0B39	. . .	; BEFORE WHICH NEW LINE IS *	
3541	0B39	. . .	; TO BE INSERTED *	
3542	0B39	. . .	; A,B=NEXT PAGE FIELD ADDR OF LINE *	
3543	0B39	. . .	; TO BE INSERTED	
3544	0B39	. . .	; EXIT : C,B = A,B(ENTRY)	
3545	0B39	. . .	; D=L DESTROYED	
3546	0B39	. . .	;*****	
3547	0B39	. . .	LININA EQU \$	
3548	0B39	7B . .	MOV A,E ;PUT ROLLED LINE ADDRESS INT	
3549	0B3A	42 . .	MOV B,D ;B,A	
3550	0B3B	EB . .	XCHG ;PUT CHAR ADDRESS INTO D,E	
3551	0B3C	. . .	LININ1 EQU \$	
3552	0B3C	6B . .	MOV L,E ;UPDATE PREV LINE PTR	
3553	0B3D	62 . .	MOV H,D ;IN NEXT LINE	
3554	0B3E	23 . .	INX H ;SET ADDRESS TO PREVIOUS	
3555	0B3F	23 . .	INX H ;LINE POINTER	
3556	0B40	4E . .	MOV C,M ;GET ADDR OF PREV LINE	
3557	0B41	77 . .	MOV M,A ;STORE ADDR OF NEW LINE	
3558	0B42	23 . .	INX H	
3559	0B43	56 . .	MOV D,M	
3560	0B44	70 . .	MOV M,B	
3561	0B45	. . .	;*****	
3562	0B45	. . .	; UPDATE NEXT/PREVIOUS POINTERS *	
3563	0B45	. . .	; IN NEW LINE *	
3564	0B45	. . .	;*****	
3565	0B45	6F . .	MOV L,A ;GET ADDR OF NEXT LINE FIELD	
3566	0B46	7C . .	MOV A,H	
3567	0B47	60 . .	MOV H,B	
3568	0B48	1D . .	DCR E ;SKIP OVER POINTERS	
3569	0B49	73 . .	MOV M,E ;STORE NEXT LINE LSB'S	
3570	0B4A	23 . .	INX H	
3571	0B4B	77 . .	MOV M,A ;STORE NEXT LINE MSB'S	
3572	0B4C	23 . .	INX H	
3573	0B4D	71 . .	MOV M,C ;STORE PREV LINE LSB'S	
3574	0B4E	23 . .	INX H	
3575	0B4F	72 . .	MOV M,D ;STORE PREV LINE MSB'S	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 105
3577	0B50	.	.	*****	
3578	0B50	.	.	; SEE IF NEW LINE IS FIRST LINE *	
3579	0B50	.	.	*****	
3580	0B50	79	.	MOV A,C ;GET PREV LINE LSB'S	
3581	0B51	B7	.	ORA A ;SET FLAGS	
3582	0B52	7D	.	MOV A,L ;(PUT LSB OF ADDR IN A-REG)	
3583	0B53	CA	61 0B	JZ LII200 ;JUMP IF NEW LINE IS	
3584	0B56	.	.	; FIRST LINE	
3585	0B56	.	.	*****	
3586	0B56	.	.	; NEW LINE IS NOT FIRST LINE *	
3587	0B56	.	.	; LINK PREVIOUS LINE TO NEW LINE *	
3588	0B56	.	.	*****	
3589	0B56	D6	04 .	SUI 4 ;GET ADDR OF NEW LINE DATA	
3590	0B58	69	.	MOV L,C ;GET ADDR OF NEXT PAGE FIELD	
3591	0B59	62	.	MOV H,D ;OF PREVIOUS LINE	
3592	0B5A	4F	.	MOV C,A ;NEW LINE'S LSB TO C	
3593	0B5B	23	.	INX H ;SET TO MSB PART OF FIELD	
3594	0B5C	CD	CA 10	CALL DISLNK ;LINK PREV LINE TO NEW LINE	
3595	0B5F	0C	.	INR C	
3596	0B60	C9	.	RET ;RETURN	
3597	0B61	.	.	*****	
3598	0B61	.	.	; NEW LINE IS FIRST LINE *	
3599	0B61	.	.	*****	
3600	0B61	.	.	LII200 EQU \$	
3601	0B61	D6	03 .	SUI 3 ;GET ADDR OF NEXT PAGE FIELD	
3602	0B63	4F	.	MOV C,A ;PUT LSB INTO C-REGISTER	
3603	0B64	6F	.	MOV L,A ;SET NEW FIRST LINE POINTER	
3604	0B65	22	9F FF	SHLD FLINE	
3605	0B68	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 106
=====
3607      0B69      . . .      ;*****
3608      0B69      . . .      ; LINE FEED PROCESSOR *
3609      0B69      . . .      ;*****
3610      0B69      . . .      CUNDLF EQU $
3611      0B69      3A FB FF      LDA  KBJMPR      ;GET THE STRAP SETTINGS
3612      0B6C      E6 04 .      ANI  LINWRP      ;WRAP AROUND ENABLED?
3613      0B6E      C8 . .      RZ              ;YES - LF NOT REQUIRED
3614      0B6F      . . .      LNFEED EQU $
3615      0B6F      . . .      ;***** GRAPHICS MODIFICATION *****
3616      0B6F      3A 97 90      LDA  ZGFLG6      ;GRAPHICS TEXT MODE? OR
3617      0B72      E6 02 .      ANI  GTEXT
3618      0B74      C4 4A 60      CNZ  ZLF          ;PROCESS IN A/N?
3619      0B77      D8 . .      RC              ;NO, PROCESS IN GRAPHICS
3620      0B78      . . .      ;*****
3621      0B78      21 6C FF      LXI  H,SPOWL      ;CLEAR SPOW LATCH
3622      0B7B      36 FF .      MVI  M,SPOWOF
3623      0B7D      2E C0 .      MVI  L,CURROW-BASE ;GET CURSOR ROW
3624      0B7F      7E . .      MOV  A,M
3625      0B80      FE 17 .      CPI  MAXROW      ;IS CURSOR IN BOTTOM ROW?
3626      0B82      CA 8A 0B      JZ   LNF100      ;YES - ROLL UP THE DISPLAY
3627      0B85      3C . .      INR  A           ;NO - MOVE CURSOR TO NEXT RO
3628      0B86      77 . .      MOV  M,A         ;STORE NEW ROW NUMBER
3629      0B87      . . .      ;***** GRAPHICS MODIFICATION *****
3630      0B87      CD 0B 60      CALL ZANCHK      ;SET SCREEN CURSOR
3631      0B8A      . . .      ;*****
3632      0B8A      . . .      LNF100 EQU $
3633      0B8A      CC 30 0D      CZ   ROLLUP      ;(ROLL UP IF AT BOTTOM)
3634      0B8D      . . .      ;
3635      0B8D      . . .      ; BUILD FIRST BLOCK OF NEW ROW IF NECESSARY
3636      0B8D      . . .      ;
3637      0B8D      3A 70 FF      LDA  MFLGS      ;GET BLOCK XFR PENDING FLAGS
3638      0B90      E6 40 .      ANI  SENTER/256 ;ENTER PENDING?
3639      0B92      C0 . .      RNZ .          ;YES - DO NOT BUILD NEW ROW
3640      0B93      3A 64 FF      LDA  IOFLG2      ;NO - GET I/O FLAGS
3641      0B96      E6 20 .      ANI  XDS2BF      ;DISPLAY TO I/O BUFFER?
3642      0B98      C0 . .      RNZ              ;YES - DO NOT BUILD NEW ROW
3643      0B99      . . .      ;
3644      0B99      . . .      ; ACQUIRE MEMORY FOR EDIT MODE IF NEEDED
3645      0B99      . . .      ;
3646      0B99      3E FF .      MVI  A,-1        ;LOCATE BEGINNING OF NEW
3647      0B9B      CD 10 0B      CALL RCADRO      ;ROW
3648      0B9E      CD 78 11      CALL CKEDIT      ;CHECK FOR SUFFICIENT FREE
3649      0BA1      C4 5C 16      CNZ  FRECNT
3650      0BA4      C9 . .      RET              ;RETURN
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 107
3652	0BA5	.	.	.	;
3653	0BA5	.	.	.	; * * * * *
3654	0BA5	.	.	.	;
3655	0BA5	.	.	.	; LSTLUP - UPDATE "LSTLIN"
3656	0BA5	.	.	.	;
3657	0BA5	.	.	.	; ENTRY: H,L = ADDRESS TO BE STORED
3658	0BA5	.	.	.	;
3659	0BA5	.	.	.	; EXIT : D,E = LSTLIN = H,L(ENTRY)
3660	0BA5	.	.	.	; A,H,L DESTROYED
3661	0BA5	.	.	.	; LSTDCD = 0
3662	0BA5	.	.	.	; PROFLD SET TO INDICATE PROTECTED
3663	0BA5	.	.	.	; FIELD OF FORMAT MODE ENABLED
3664	0BA5	.	.	.	;
3665	0BA5	.	.	.	LSTLUP EQU \$
3666	0BA5	22	C9	FF	SHLD LSTLIN ;SET NEW "LSTLIN" VALUE
3667	0BA8	.	.	.	LSTLU1 EQU \$
3668	0BA8	EB	.	.	XCHG ;PUT "LSTLIN" VALUE INTO D,E
3669	0BA9	.	.	.	LSTLU2 EQU \$
3670	0BA9	AF	.	.	XRA A ;CLEAR LAST DISPLAY CODE
3671	0BAA	32	C6	FF	STA LSTDCD
3672	0BAD	3E	C0	.	MVI A,STPR ;INITIALIZE LAST FORMAT
3673	0BAF	32	C5	FF	STA LSTFMT ;CONTROL CODE TO "STPR"
3674	0BB2	CD	CF	1A	CALL CHKFMS ;FORMAT MODE?
3675	0BB5	C8	.	.	RZ ;NO - RETURN
3676	0BB6	3E	FF	.	MVI A,-1 ;YES - SET PROTECT FLAG TO
3677	0BB8	32	C2	FF	STA PROFLD ;INDICATE PROTECTED FIELD
3678	0BBB	21	0B	08	LXI H,ZRETRN ;INITIALIZE FIELD CHECKING
3679	0BBE	22	86	FF	SHLD CHKRTN ;ROUTINE
3680	0BC1	C9	.	.	RET

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 108
=====
3682 0BC2 . . . ;*****
3683 0BC2 . . . ; MEMORY LOCK OFF *
3684 0BC2 . . . ;*****
3685 0BC2 . . . MLKOF0 EQU $
3686 0BC2 3A 6B FF LDA MLKROW ;GET MEMORY LOCK ROW
3687 0BC5 87 . . ORA A ;SET FOR FULL LOCK OUT?
3688 0BC6 C2 F1 11 JNZ MLKOF ;NO - CLEAR LOCK OUT ONLY
3689 0BC9 . . . MLKOFF EQU $ ;YES - TURN OFF MEMORY LOCK
3690 0BC9 21 00 00 LXI H,0 ;SET MEMORY LOCK ROW AND
3691 0BCC 22 6A FF SHLD MLKFLG ;FLAG TO ZERO
3692 0BCF 3E 04 . MVI A, MEMLOK ;TURN OFF MEMORY LOCK
3693 0BD1 C3 11 48 JMP ZCLMD1 ;FLAG
3694 0BD4 . . . ;*****
3695 0BD4 . . . ; MEMORY LOCK ON *
3696 0BD4 . . . ;*****
3697 0BD4 . . . MLKUN EQU $
3698 0BD4 3A C0 FF LDA CURROW ;GET CURRENT CURSOR ROW
3699 0BD7 87 . . ORA A ;SET FOR OVERFLOW INHIBIT?
3700 0BD8 C2 DF 0B JNZ MLO005 ;NO - SET MEMORY LOCK ROW
3701 0BD8 CD 78 11 CALL CKEDIT ;EDIT MODE?
3702 0BDE C0 . . RNZ ;YES - DON'T ALLOW LOCK OUT
3703 0BDF . . . MLO005 EQU $ ;NO - SET MEMORY LOCK ROW
3704 0BDF 32 6B FF STA MLKROW
3705 0BE2 . . . MLO010 EQU $
3706 0BE2 3E 04 . MVI A, MEMLOK ;TURN MEMORY LOCK FLAG
3707 0BE4 06 00 . MVI B,0 ;ON AND DON'T BLINK LED
3708 0BE6 21 6A FF LXI H, MLKFLG ;(CLEAR THE MEMORY LOCK
3709 0BE9 70 . . MOV M, B ;FLAG)
3710 0BEA C3 0E 48 JMP ZSTMD1
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 109
3712	OBED	.	.	*****	
3713	OBED	.	.	; MLKSCH - LOCATE MEMORY LOCK ROW *	
3714	OBED	.	.	*****	
3715	OBED	.	.	;	
3716	OBED	.	.	; ENTRY: DON'T CARE	
3717	OBED	.	.	;	
3718	OBED	.	.	; EXIT : Z - MEMORY LOCK ROW NOT FOUND	
3719	OBED	.	.	; A,C,H,L DESTROYED	
3720	OBED	.	.	; NZ - MEMORY LOCK ROW FOUND	
3721	OBED	.	.	; H,L = ADDRESS OF LAST LOCK ROW	
3722	OBED	.	.	; (POINTS TO LSB OF NEXT LINE	
3723	OBED	.	.	; POINTER)	
3724	OBED	.	.	; A,C DESTROYED	
3725	OBED	.	.	;	
3726	OBED	.	.	MLKSCO EQU \$;LOCATE FIRST UNLOCKED ROW	
3727	OBED	3A	6B FF	LDA MLKROW ;GET MEMORY LOCK ROW	
3728	0BF0	B7	.	ORA A ;SET FOR PARTIAL SCREEN LOCK	
3729	0BF1	2A	CB FF	LHLD TOPLIN ;(SET FOR TOP DISPLAY LINE	
3730	0BF4	CA	0A 0C	JZ NZEXIT ;NO - RETURN FOUND (NZ)	
3731	0BF7	.	.	; YES - LOCATE MEMORY LOCK ROW	
3732	0BF7	.	.	MLKSCH EQU \$	
3733	0BF7	3A	6B FF	LDA MLKROW ;GET MEMORY LOCK ROW	
3734	0BFA	B7	.	ORA A ;SET FOR PARTIAL SCREEN LOCK	
3735	0BFB	C8	.	RZ ;NO - RETURN	
3736	0BFC	.	.	*****	
3737	0BFC	.	.	; SEARCH FOR ROW *	
3738	0BFC	.	.	*****	
3739	0BFC	2A	CB FF	LHLD TOPLIN ;GET TOP LINE ADDRESS	
3740	0BFF	.	.	MLKSC1 EQU \$;LOCATE LINE (A-REG)	
3741	0BFF	4F	.	MOV C,A ;PUT LINE NUMBER IN C-REG	
3742	0C00	.	.	MLS120 EQU \$	
3743	0C00	CD	C6 1A	CALL CHAIN ;GET ADDRESS OF NEXT LINE	
3744	0C03	B7	.	ORA A ;DOES NEXT LINE EXIST?	
3745	0C04	C8	.	RZ ;NO - RETURN FAIL (Z)	
3746	0C05	23	.	INX H ;YES - SET TO NEXT LINE PTR	
3747	0C06	0D	.	DCR C ;ALL LINES FOUND?	
3748	0C07	C2	00 0C	JNZ MLS120 ;NO - DO NEXT LINE	
3749	0C0A	.	.	;	
3750	0C0A	.	.	NZEXIT EQU \$	
3751	0C0A	F6	FF	ORI 3770 ;SET NZ, S	
3752	0C0C	C9	.	RET ;RETURN(ZERO FLAG FALSE)	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 110
=====
3754      0C0D      . . .      ;*****
3755      0C0D      . . .      ; MLOCK - TURN ON MEMORY LOCK FULL CONDITION *
3756      0C0D      . . .      ;*****
3757      0C0D      . . .      ;
3758      0C0D      . . .      ; ENTRY:  DON'T CARE
3759      0C0D      . . .      ;
3760      0C0D      . . .      ; EXIT :  A = 0
3761      0C0D      . . .      ;          Z = T
3762      0C0D      . . .      ;          MLKTMR = -1 (377B)
3763      0C0D      . . .      ;
3764      0C0D      . . .      MLOCK0 EQU $
3765      0C0D      CD  44  07      CALL PTB100      ;RESTORE PROPER DISPLAY PARM
3766      0C10      . . .      MLOCK EQU $
3767      0C10      21  6A  FF      LXI H,MLKFLG    ;SET H,L TO MEMORY LOCK FLAG
3768      0C13      86 . .      URA M           ;MEMORY ALREADY LOCKED?
3769      0C14      C2  22  0C      JNZ MLK010      ;YES - DON'T SOUND BELL
3770      0C17      3E  04 .      MVI A,MEMLOK   ;NO - FORCE MEMORY LOCK ON
3771      0C19      06  FF .      MVI B,377Q     ;AND BLINKING
3772      0C1B      70 . .      MOV M,B        ;SET MEMORY LOCK FLAG
3773      0C1C      CD  0E  48      CALL ZSTMD1
3774      0C1F      . . .      MLOCK1 EQU $    ;SOUND BELL AND RETURN A = 0
3775      0C1F      CD  14  48      CALL ZBELL     ;SOUND THE BELL
3776      0C22      . . .      MLK010 EQU $
3777      0C22      AF . .      XRA A          ;SET Z-FLAG
3778      0C23      21  9A  FF      LXI H,NROWS    ;(SET H TO DATA PAGE)
3779      0C26      77 . .      MOV M,A        ;CLEAR NROWS FOR RCADDR
3780      0C27      C9 . .      RET           ;RETURN (A = 0, Z= T)
3781      0C28      00 . .      NOP          ;NOP FOR PATCH TO "PT772"
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 111
3783	0C29	. . .	;	
3784	0C29	. . .	; * * * * *	
3785	0C29	. . .	;	
3786	0C29	. . .	; MOVCHR - MOVE CHARACTER STRING	
3787	0C29	. . .	;	
3788	0C29	. . .	; ENTRY: H,L = SOURCE POINTER	
3789	0C29	. . .	; B,C = DESTINATION POINTER	
3790	0C29	. . .	;	
3791	0C29	. . .	; EXIT : B,C = NEXT STORAGE LOCATION	
3792	0C29	. . .	; H,L = END OF SOURCE STRING	
3793	0C29	. . .	; Z - TERMINATED BY A NULL BYTE	
3794	0C29	. . .	; NZ - TERMINATED BY AN EOP	
3795	0C29	. . .	;	
3796	0C29	. . .	MOVCHR EQU \$	
3797	0C29	7E . .	MOV A,M ;GET DATA BYTE	
3798	0C2A	B7 . .	ORA A ;IS IT A NULL?	
3799	0C2B	C8 . .	RZ ;YES - RETURN (Z - TRUE)	
3800	0C2C	02 . .	STAX B ;NO - STORE THE BYTE	
3801	0C2D	23 . .	INX H ;INCREMENT TO NEXT SOURCE BY	
3802	0C2E	0B . .	DCX B ;DECREMENT TO NEXT DEST BYTE	
3803	0C2F	FE CE .	CPI EOP ;WAS LAST BYTE AN EOP?	
3804	0C31	C2 29 0C	JNZ MOVCHR ;NO - DO NEXT BYTE	
3805	0C34	B7 . .	ORA A ;YES - SET Z-FALSE	
3806	0C35	C9 . .	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 112
=====
3808      0C36      .      .      .      ;*****
3809      0C36      .      .      .      ; NEXT PAGE *
3810      0C36      .      .      .      ;*****
3811      0C36      .      .      .      NEXTPG EQU $
3812      0C36      3E      18      .      MVI  A,MAXROW+1  ;COMPUTE NUMBER OF LINES
3813      0C38      2E      6B      .      MVI  L,MLKROW    ;TO ROLL UP
3814      0C3A      96      .      .      SUB  M
3815      0C38      CD      4E      0C      CALL NXT100
3816      0C3E      .      .      .      NXT040 EQU $
3817      0C3E      3A      6B      FF      LDA  MLKROW      ;SET CURRENT CURSOR POSITION
3818      0C41      32      C0      FF      STA  CURROW      ;TO MEMORY LOCK ROW AND
3819      0C44      CD      7C      23      CALL CURPRT      ;LEFT MARGIN
3820      0C47      CD      CF      1A      CALL CHKFMS      ;FORMAT/SOFT KEY DEFINE MODE
3821      0C4A      C2      3A      1F      JNZ  FLDSR       ;YES - TAB TO NEXT FIELD
3822      0C4D      C9      .      .      RET              ;NO - RETURN
=====

```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 114
3861	0C67	. . .	;*****	
3862	0C67	. . .	; GET ADDRESS OF NEXT	*
3863	0C67	. . .	; RAM BLOCK.	*
3864	0C67	. . .	; ENTRY	*
3865	0C67	. . .	; E, BIT 7 = 0, 4K INCREMENTS	*
3866	0C67	. . .	; = 1, 256	*
3867	0C67	. . .	; BIT 0 = 0, IN NON-DISPLAY RAM	*
3868	0C67	. . .	; = 1, DISPLAY RAM	*
3870	0C67	. . .	; H = 0 IF FIRST ENTRY OF ROUTINE	*
3871	0C67	. . .	; CALL NXSBLK	*
3872	0C67	. . .	; EXIT	*
3873	0C67	. . .	; (H,L) = ADDRESS OF NEXT	*
3874	0C67	. . .	; BLOCK	*
3875	0C67	. . .	; A = 0 IF END OF MEMORY	*
3876	0C67	. . .	; E SET TO INDICATE APPROP. RAM	*
3877	0C67	. . .	; OTHER REGS. UNCHANGED, FLAGS ARE.	*
3878	0C67	. . .	;*****	
3879	0C67	. . .	NXSBLK EQU \$	
3880	0C67	. . .	PUSH B	
3881	0C67	. . .	XRA A	
3882	0C68	AF . .	CMP H ;H = 0?	
3883	0C69	BC . .	JNZ NXB100 ;NO - ADVANCE TO NEXT BLOCK	
3884	0C6A	C2 7E 0C	LHLD BUFBN ;IS THERE ANY NON DISPLAY	
3885	0C6D	2A 8D FF	LHLD NXB060 EQU \$	
3886	0C70	. . .	LDA BUFEND+1 ;MEMORY?	
3887	0C70	3A 8C FF	CMP H	
3888	0C73	BC . .	JNC NXB200 ;YES, EXIT	
3889	0C74	D2 8C 0C	LHLD DSPBN ;NO, USE DISPLAY MEMORY	
3890	0C77	2A AA FF	INR E ;INDICATE DISPLAY MEMORY	
3891	0C7A	1C . .	JMP NXB200 ;EXIT	
3892	0C7B	C3 8C 0C	NXB100 EQU \$	
3893	0C7E	. . .	ORA E ;INCREMENT BY 4K (BIT 7 = 0)	
3894	0C7E	B3 . .	LXI B,10000Q ;(SET FOR 4K INCREMENT)	
3895	0C7F	01 00 10	JP NXB150 ;YES - COMPUTE NEXT BLOCK AD	
3896	0C82	F2 87 0C	MVI B,256/256 ;NO - INCREMENT BY 256 ONLY	
3897	0C85	06 01 .	NXB150 EQU \$	
3898	0C87	. . .	DAU B ;BUMP POINTER	
3899	0C87	09 . .	RRC ;TESTING NON-DISPLAY AREA?	
3900	0C88	0F . .	JNC NXB060 ;YES - CHECK UPPER BOUNDARY	
3901	0C89	D2 70 0C	NXB200 EQU \$	
3902	0C8C	. . .	MOV A,H ;IF WE WENT OVER TOP OF	
3903	0C8C	7C . .	; MEMORY H, = 0	
3904	0C8D	. . .	POP B	
3905	0C8D	C1 . .	RET	
3906	0C8E	C9 . .		

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 115
3910	0C8F	.	.	*****	
3911	0C8F	.	.	; NXTCHR - GET NEXT CHARACTER IN DISPLAY LIST *	
3912	0C8F	.	.	*****	
3913	0C8F	.	.	;	
3914	0C8F	.	.	; ENTRY: D,E = ADDRESS OF CURRENT CHARACTER	
3915	0C8F	.	.	;	
3916	0C8F	.	.	; EXIT : Z = T, CHARACTER IS NOT AN EOL LINK	
3917	0C8F	.	.	A = DISPLAY CHARACTER	
3918	0C8F	.	.	D,E = ADDRESS OF CHARACTER	
3919	0C8F	.	.	F, NEXT CHARACTER IS EOL LINK	
3920	0C8F	.	.	A DESTROYED	
3921	0C8F	.	.	D,E = ADDRESS OF NEXT LINE LINK	
3922	0C8F	.	.	;	
3923	0C8F	.	.	NXTCH0 EQU \$	
3924	0C8F	EB	.	XCHG ;PUT POINTER INTO D,E	
3925	0C90	.	.	NXTCHR EQU \$	
3926	0C90	1B	.	DCX D ;GET THE NEXT DISPLAY	
3927	0C91	1A	.	LDAX D ;CHARACTER	
3928	0C92	FE	D0	CPI LNKLIM ;IS IT A LINK?	
3929	0C94	DA	A2 0C	JC NCH010 ;NO - EXIT	
3930	0C97	EB	.	XCHG ;YES - GET NEW ADDRESS	
3931	0C98	2B	.	DCX H ;GET LSB OF LINK	
3932	0C99	6E	.	MOV L,M	
3933	0C9A	67	.	MOV H,A	
3934	0C9B	EB	.	XCHG ;PUT ADDRESS INTO D,E	
3935	0C9C	7B	.	MOV A,E ;PUT LSB INTO A-REGISTER	
3936	0C9D	2F	.	CMA ;END OF LINE LINK (LOWER FOU	
3937	0C9E	E6	0F	ANI BLKSM ;BITS NOT ALL ONES)?	
3938	0CA0	C0	.	RNZ ;YES - RETURN Z FALSE	
3939	0CA1	1A	.	LDAX D ;NO - GET THE DATA BYTE	
3940	0CA2	.	.	;	
3941	0CA2	.	.	NCH010 EQU \$	
3942	0CA2	BF	.	CMP A ;SET Z TRUE	
3943	0CA3	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 116
=====
3945      OCA4      . . .      ;*****
3946      OCA4      . . .      ; PAROUT - SEND STATUS BITS *
3947      OCA4      . . .      ;*****
3948      OCA4      . . .      ;
3949      OCA4      . . .      ; ENTRY:  A = PARITY BITS TO BE SENT
3950      OCA4      . . .      ;
3951      OCA4      . . .      ; EXIT :  A-E DESTROYED
3952      OCA4      . . .      ;
3953      OCA4      . . .      PAROT4 EQU $          ;ROTATE DOWN 4 BITS FIRST
3954      OCA4      0F . . .      RRC
3955      OCA5      . . .      PAROT3 EQU $
3956      OCA5      0F . . .      RRC
3957      OCA6      . . .      PAROT2 EQU $
3958      OCA6      0F . . .      RRC
3959      OCA7      . . .      PAROT1 EQU $
3960      OCA7      0F . . .      RRC
3961      OCA8      . . .      PAROUT EQU $
3962      OCA8      E6 0F .      ANI 170             ;GET BITS 0-3
3963      OCAA     C6 30 .      ADI ZERO           ;ADD IN ZERO BASE TO FORCE
3964      OCAC     E5 . . .      PUSH H             ;DISPLAYABLE CHARACTER
3965      OCAD     CD CD FF  CALL ECONTF        ;PERFORM OUTPUT FUNCTION
3966      OCB0     E1 . . .      POP H              ;RESTORE H,L
3967      OCB1     C9 . . .      RET                ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 117
3969	OCB2	.	.	. ;*****	
3970	OCB2	.	.	. ; PREVIOUS PAGE *	
3971	OCB2	.	.	. ;*****	
3972	OCB2	.	.	. PREVPG EQU \$	
3973	OCB2	3E	E8	. MVI A,-MAXROW-1	
3974	OCB4	2E	6B	. MVI L,MLKROW ;COMPUTE NUMBER OF ROWS TO	
3975	OCB6	86	.	. ADD M ;ROLL DOWN	
3976	OCB7	CD	BF	0C CALL PRV100	
3977	OCBA	C3	3E	0C JMP NXT040	
3978	OCBD	.	.	. ;	
3979	OCBD	.	.	. ; PRVPG1 - ROLL DOWN FOR CURSOR POSITIONING	
3980	OCBD	.	.	. ;	
3981	OCBD	.	.	. ; ENTRY: H,L = CURROW+	
3982	OCBD	.	.	. ;	
3983	OCBD	.	.	. PRVPG1 EQU \$	
3984	OCBD	36	00	. MVI M,0 ;SET CURRENT ROW TO ZERO	
3985	OCBF	.	.	. ;	
3986	OCBF	.	.	. ; * * * * *	
3987	OCBF	.	.	. ;	
3988	OCBF	.	.	. ; PRV100 - ROLL DOWN N LINES	
3989	OCBF	.	.	. ;	
3990	OCBF	.	.	. ; ENTRY: A = -NUMBER OF LINES TO ROLL DOWN	
3991	OCBF	.	.	. ; H = BASEH	
3992	OCBF	.	.	. ;	
3993	OCBF	.	.	. ; EXIT : A-L DESTROYED	
3994	OCBF	.	.	. ;	
3995	OCBF	.	.	. ;	
3996	OCBF	.	.	. PRV100 EQU \$	
3997	OCBF	32	82	FF STA ROLLCT ;SAVE THE ROLL COUNT	
3998	OCC2	.	.	. PRV110 EQU \$	
3999	OCC2	CD	CE	0C CALL ROLLDN ;LINE ROLLED DOWN?	
4000	OCC5	21	82	FF LXI H,ROLLCT ;(SET H TO DATA PAGE)	
4001	OCC8	C8	.	. RZ ;NO - RETURN	
4002	OCC9	34	.	. INR M ;ALL LINES DONE?	
4003	OCCA	C2	C2	0C JNZ PRV110 ;NO - DO ANOTHER LINE	
4004	OCDD	C9	.	. RET ;YES - RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 118
=====
4006      OCCE      . . .      ;*****
4007      OCCE      . . .      ; ROLLDN - ROLL DISPLAY DOWN ONE LINE *
4008      OCCE      . . .      ;*****
4009      OCCE      . . .      ;
4010      OCCE      . . .      ; ENTRY:  DON'T CARE
4011      OCCE      . . .      ;
4012      OCCE      . . .      ; EXIT :  NZ - ROLL DOWN SUCCESSFUL
4013      OCCE      . . .      ; Z - ROLL DOWN FAILED
4014      OCCE      . . .      ; ALL REGISTERS DESTROYED
4015      OCCE      . . .      ;
4016      OCCE      . . .      ROLLDN EQU $
4017      OCCE      CD F7 0B      CALL MLKSCH
4018      OCD1      CA F8 0C      JZ RLD080
4019      OCD4      . . .      ;*****
4020      OCD4      . . .      ; MEMORY LOCK ROLL DOWN *
4021      OCD4      . . .      ;*****
4022      OCD4      EB . .      XCHG ;LAST LOCKED LINE ADDR TO D,
4023      OCD5      2A CB FF      LHL D TOPLIN ;GET TOP LINE ADDRESS
4024      OCD8      23 . .      INX H ;SET ADDRESS TO PREVIOUS LIN
4025      OCD9      23 . .      INX H ;POINTER
4026      OCDA      CD C6 1A      CALL CHAIN ;GET PREVIOUS LINE'S ADDRESS
4027      OCDD      B7 . .      ORA A ;PREVIOUS LINE EXIST?
4028      OCDE      C8 . .      RZ ;NO - RETURN
4029      OCDF      D5 . .      PUSH D ;YES - ROLL DOWN THE LINE
4030      OCE0      CD DA 0A      CALL LINDLO ;DELETE 1ST LINE ABOVE DISP
4031      OCE3      21 A3 FF      LXI H,TLINO ;DECREMENT TOP LINE
4032      OCE6      35 . .      DCR M ;NUMBER
4033      OCE7      E1 . .      POP H ;RECALL LAST LOCKED LINE ADD
4034      OCE8      CD 39 0B      CALL LININA ;ADD LINE BELOW LOCKED LINES
4035      OCEB      3A 6B FF      LDA MLKROW ;GET LOCK ROW NUMBER
4036      OCEE      3D . .      DCR A ;ADJUST FOR COMPARE
4037      OCEF      21 C7 FF      LXI H,LSTROW ;COMPARE TO LAST ROW DONE
4038      OCF2      BE . .      CMP M ;DID IT ROLL DOWN?
4039      OCF3      FA 21 0D      JM RLD090 ;YES - UPDATE DISPLAY PTRS
4040      OCF6      B4 . .      ORA H ;NO - FORCE NZ AND EXIT
4041      OCF7      C9 . .      RET ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 119
4043	0CF8	.	.	. ;*****	
4044	0CF8	.	.	. ; NORMAL ROLL DOWN *	
4045	0CF8	.	.	. ;*****	
4046	0CF8	.	.	. RLD080 EQU \$	
4047	0CF8	3A	6B	FF LDA MLKROW ;GET MEMORY LOCK ROW	
4048	0CF8	B7	.	. ORA A ;IS IT ZERO?	
4049	0CFC	CA	13	0D JZ RLD085 ;YES - DO NORMAL ROLL DOWN	
4050	0CFF	21	C0	FF LXI H,CURROW ;NO - TRY TO ALLOCATE LINES	
4051	0D02	46	.	. MOV B,M ;TO MEMORY LOCK ROW	
4052	0D03	77	.	. MOV M,A	
4053	0D04	C5	.	. PUSH B ;SAVE CURRENT ROW NUMBER	
4054	0D05	3E	FF	. MVI A,-1 ;(SET FOR COLUMN ZERO)	
4055	0D07	CD	10	08 CALL RCADRO ;IS MEMORY AVAILABLE?	
4056	0D0A	C1	.	. POP B ;(RESTORE CURRENT ROW	
4057	0D0B	78	.	. MOV A,B ;NUMBER)	
4058	0D0C	32	C0	FF STA CURROW	
4059	0D0F	C0	.	. RNZ ;NO - RETURN FAIL	
4060	0D10	C3	CE	0C JMP ROLLDN ;YES - RETRY MEMORY LOCK ROL	
4061	0D13	.	.	. ;	
4062	0D13	.	.	. ; DISPLAY NOT LOCKED - DO NORMAL ROLL DOWN	
4063	0D13	.	.	. ;	
4064	0D13	.	.	. RLD085 EQU \$	
4065	0D13	2A	CB	FF LHLD TOPLIN ;GET TOP LINE ADDRESS	
4066	0D16	23	.	. INX H ;SET TO PREVIOUS LINE	
4067	0D17	23	.	. INX H ;ADDRESS	
4068	0D18	B6	.	. ORA M ;ANY PREVIOUS LINES?	
4069	0D19	C8	.	. RZ ;NO - DON'T DO ROLL DOWN	
4070	0D1A	.	.	. ; YES - ROLL ONE LINE DOWN	
4071	0D1A	.	.	. ;*****	
4072	0D1A	.	.	. ; TOP LINE IS NOT FIRST LINE *	
4073	0D1A	.	.	. ; ADVANCE POINTERS *	
4074	0D1A	.	.	. ;*****	
4075	0D1A	16	FF	. MVI D,-1 ;FLAG TO DECREMENT TLIN0	
4076	0D1C	CD	A3	10 CALL TOPUPD ;UPDATE TOP LINE POINTERS	
4077	0D1F	2E	C7	. MVI L,LSTROW-BASE ;GET LAST ROW PROCESSED	
4078	0D21	.	.	. RLD090 EQU \$	
4079	0D21	7E	.	. MOV A,M	
4080	0D22	3C	.	. INK A ;INCREMENT	
4081	0D23	FE	18	. CPI MAXROW+1	
4082	0D25	C2	57	17 JNZ STOREA ;NOT ROLL OFF - STORE ROW	
4083	0D28	2A	C9	FF LHLD LSTLIN ;GET ADDR OF LAST LINE DONE	
4084	0D2B	23	.	. INX H ;SET TO PREVIOUS LINE	
4085	0D2C	23	.	. INX H ;ADDRESS	
4086	0D2D	C3	5C	0D JMP ROL200	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 120
=====
4088      0D30      . . .      ;*****
4089      0D30      . . .      ; ROLLUP - ROLL UP DISPLAY ONE LINE *
4090      0D30      . . .      ;*****
4091      0D30      . . .      ROLLUP EQU $
4092      0D30      CD F7 0B      CALL MLKSCH
4093      0D33      CA 6F 0D      JZ ROL080
4094      0D36      . . .      ;*****
4095      0D36      . . .      ; MEMORY LOCK ROLL-UP *
4096      0D36      . . .      ;*****
4097      0D36      7E . .      MOV A,M      ;IS THERE A NEXT LINE?
4098      0D37      87 . .      ORA A
4099      0D38      C8 . .      RZ      ;NO - DON'T DO ROLL UP
4100      0D39      CD DA 0A      CALL LINDLO  ;YES - REMOVE FIRST UNLOCKED
4101      0D3C      21 A3 FF      LXI H,TLINO  ;LINE
4102      0D3F      34 . .      INR M      ;INCREMENT TOP LINE NUMBER
4103      0D40      2A CB FF      LHLD TOPLIN  ;GET TOP DISPLAY LINE ADDRES
4104      0D43      3A 6B FF      LDA MLKROW   ;FORCE END-OF-PAGE IF DISPLA
4105      0D46      F6 20 .      ORI MAYEOP   ;IS CURRENTLY REFRESHING
4106      0D48      . . .      ;***** GRAPHICS MODIFICATION *****
4107      0D48      CD 0B 60      CALL ZANCHK  ;MEMORY LOCK BOUNDRY ROW
4108      0D4B      . . .      ;*****
4109      0D4B      CD 39 0B      CALL LININA  ;ADD LINE ABOVE DISPLAY
4110      0D4E      3A 6B FF      LDA MLKROW  ;GET LOCK ROW NUMBER
4111      0D51      21 C7 FF      LXI H,LSTROW ;GET LAST ROW PROCESSED
4112      0D54      96 . .      SUB M      ;DID IT ROLL UP?
4113      0D55      FA 7D 0D      JM ROL090   ;YES - UPDATE LINE POINTER
4114      0D58      C0 . .      RNZ      ;NO - RETURN (Z = FALSE)
4115      0D59      77 . .      MOV M,A    ;SAME - FORCE LAST ROW = 0
4116      0D5A      . . .      ROL100 EQU $
4117      0D5A      2E CB .      MVI L,TOPLIN ;SET CURRENT LINE TO TOP LINE
4118      0D5C      . . .      ROL200 EQU $
4119      0D5C      5E . .      MOV E,M
4120      0D5D      . . .      ROLUP2 EQU $
4121      0D5D      2C . .      INR L
4122      0D5E      56 . .      MOV D,M
4123      0D5F      . . .      ;
4124      0D5F      . . .      ; ROLUP3 - UPDATE LSTLIN AND CURADR
4125      0D5F      . . .      ;
4126      0D5F      . . .      ROLUP3 EQU $
4127      0D5F      EB . .      XCHG      ;SET LSTLIN TO NEW ROW
4128      0D60      . . .      ROLUPC EQU $
4129      0D60      CD A5 0B      CALL LSTLUP
4130      0D63      EB . .      XCHG      ;PUT NEW ROW ADDRESS INT H,L
4131      0D64      2B . .      DCX H     ;SET TO LSB OF NEXT LINE PTR
4132      0D65      22 C3 FF      SHLD CURADR ;SET CURADR TO TOP LINE
4133      0D68      EB . .      XCHG      ;RESTURE D,E AND H,L
4134      0D69      AF . .      XRA A     ;SET LAST COLUMN PROCESSED
4135      0D6A      32 C8 FF      STA LSTCOL ;DONE TO ZERO
4136      0D6D      B3 . .      URA E     ;SET Z-FLAG FALSE
4137      0D6E      C9 . .      RET      ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 121
4139	0D6F	.	.	. ;*****	
4140	0D6F	.	.	. ; NORMAL ROLL-UP *	
4141	0D6F	.	.	. ;*****	
4142	0D6F	.	.	. ROLU80 EQU \$	
4143	0D6F	2A	CB	FF LHL D TOPLIN ;GET TOP LINE ADDRESS	
4144	0D72	B6	.	. ORA M ;IS TOP LINE LAST LINE?	
4145	0D73	C8	.	. RZ ;YES - RETURN, DON'T ROLL UP	
4146	0D74	16	01	. MVI D,1 ;NO - SET D TO INCREMENT	
4147	0D76	.	.	. ; "TLINO"	
4148	0D76	3C	.	. INR A ;SET LSB TO NEXT LINE POINTE	
4149	0D77	.	.	. ;*****	
4150	0D77	.	.	. ; TOP LINE IS NOT LAST LINE *	
4151	0D77	.	.	. ; ADVANCE POINTERS *	
4152	0D77	.	.	. ;*****	
4153	0D77	.	.	. ROLUP1 EQU \$	
4154	0D77	CD	A3	10 CALL TOPUPD ;UPDATE TOP LINE POINTERS	
4155	0D7A	21	C7	FF LXI H,LSTROW ;GET LAST ROW # PROCESSED	
4156	0D7D	.	.	. ROLU90 EQU \$	
4157	0D7D	4E	.	. MOV C,M	
4158	0D7E	0D	.	. DCR C ;DECREMENT	
4159	0D7F	FA	5A	0D JM ROL100 ;LINE ROLLED OFF SCREEN	
4160	0D82	71	.	. MOV M,C ;STORE UPDATED LSTROW	
4161	0D83	B4	.	. ORA H ;SET Z-FLAG TO FALSE	
4162	0D84	C9	.	. RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 122
=====
4164      0D85      . . .      ;*****
4165      0D85      . . .      ; CHAR SET SELECT *
4166      0D85      . . .      ;*****
4167      0D85      . . .      SCHRST EQU $
4168      0D85      21 0F 7F      LXI  H,CHRSTB  ;SET FOR CHARACTER SET SELEC
4169      0D88      C3 34 05      JMP ESCAP0
4170      0D8B      . . .      ;
4171      0D8B      . . .      ; SET NEW ALTERNATE CHARACTER SET
4172      0D8B      . . .      ;
4173      0D8B      . . .      SCHST1 EQU $
4174      0D8B      79 . .      MOV  A,C        ;PUT INPUT CHARACTER IN A-RE
4175      0D8C      E6 0F .      ANI  170        ;EXTRACT CHARACTER SET NUMBE
4176      0D8E      07 . .      RLC              ;SHIFT TO POSITION FOR
4177      0D8F      07 . .      RLC              ;ALTERNATE CHARCTER SET
4178      0D90      07 . .      RLC
4179      0D91      07 . .      RLC
4180      0D92      32 72 FF      STA  CHRSET     ;STORE CHAR SET SELECT CTL
4181      0D95      C9 . .      RET              ;RETURN
=====
  
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 123
4183	0D96	.	.	.	;
4184	0D96	.	.	.	; * * * * *
4185	0D96	.	.	.	;
4186	0D96	.	.	.	; SFKYOF - PUT NORMAL DISPLAY ON SCREEN
4187	0D96	.	.	.	;
4188	0D96	.	.	.	; ENTRY: DON'T CARE
4189	0D96	.	.	.	;
4190	0D96	.	.	.	; EXIT : ALL REGISTERS DESTROYED
4191	0D96	.	.	.	;
4192	0D96	.	.	.	; SFKYOF EQU \$
4193	0D96	CD	E5	1A	CALL CHKSFK ;NORMAL DISPLAY ENABLED?
4194	0D99	C8	.	.	RZ ;YES - RETURN
4195	0D9A	3E	F7	.	MVI A,3770-DEFSKY ;NO - SWAP DISPLAY
4196	0D9C	CD	20	15	CALL CLCMFL ;CLEAR SOFT KEY MODE FLAG
4197	0D9F	CD	72	11	CALL CKDSPF ;DISPLAY FUNCTIONS ENABLED?
4198	0DA2	C2	BA	0D	JNZ SFO010 ;YES - DON'T RESET RANGE TBL
4199	0DA5	21	FF	7D	LXI H,RTABLE ;NO - RESTORE NORMAL
4200	0DA8	22	D2	FF	SHLD RRGTA ;CHARACTER FUNCTION TABLE
4201	0DAB	C3	BA	0D	JMP SFO010 ;TURN ON NORMAL DISPLAY

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 124
4203	ODAE	.	.	*****	
4204	ODAE	.	.	; SFKYON - PUT SOFT KEY DISPLAY ON SCREEN *	
4205	ODAE	.	.	*****	
4206	ODAE	.	.	;	
4207	ODAE	.	.	; ENTRY: DON'T CARE	
4208	ODAE	.	.	;	
4209	ODAE	.	.	; EXIT : NZ	
4210	ODAE	.	.	; ALL REGISTERS DESTROYED	
4211	ODAE	.	.	;	
4212	ODAE	.	.	SFKYON EQU \$	
4213	ODAE	CD	E5 1A	CALL CHKSKF ;SOFT KEY DEFINE MODE?	
4214	ODB1	C0	.	RNZ ;YES - RETURN	
4215	ODB2	3E	08	MVI A,DEFSKY ;NO - SWAP DISPLAY	
4216	ODB4	CD	44 15	CALL STCMFL ;SET SOFT KEY MODE FLAG	
4217	ODB7	.	.	*****	
4218	ODB7	CD	50 60	CALL ZVID1 ;GRAPHICS OFF, ALLOW A/N	
4219	ODBA	.	.	*****	
4220	ODBA	.	.	;	
4221	ODBA	.	.	; EXCHANGE DISPLAY	
4222	ODBA	.	.	;	
4223	ODBA	.	.	SFO010 EQU \$	
4224	ODBA	CD	17 23	CALL SWAP ;SWAP DISPLAY PARAMETERS	
4225	ODBD	CD	6A 1E	CALL RSTDSP ;TURN ON THE DISPLAY	
4226	ODC0	CD	96 1F	CALL FLDSRX ;RESCAN LINE TO SET PROPER	
4227	ODC3	C3	A1 07	JMP RCADRA ;FIELD ATTRIBUTE	
4228	ODC6	.	.	*****	
4229	ODC6	.	.	; SFKYDS - DISPLAY CHARACTER IN SOFT KEY MODE *	
4230	ODC6	.	.	*****	
4231	ODC6	.	.	;	
4232	ODC6	.	.	; ENTRY: DCHAR = CHARACTER TO BE DISPLAYED	
4233	ODC6	.	.	;	
4234	ODC6	.	.	; EXIT : IF CHARACTER FROM KEYBOARD,	
4235	ODC6	.	.	CHARACTER IS ADDED TO DISPLAY	
4236	ODC6	.	.	OTHERWISE, NORMAL DISPLAY IS RESTORED	
4237	ODC6	.	.	;	
4238	ODC6	.	.	SFKYDS EQU \$	
4239	ODC6	.	.	*****	
4240	ODC6	CD	23 60	CALL ZMUCHK ;IS THE AUTOPLT MENU ON?	
4241	ODC9	C4	1A 60	CNZ ZAPMOF ;IF SO, TURN IT OFF	
4242	ODCC	.	.	*****	
4243	ODCC	CD	E5 1A	CALL CHKSKF ;SOFT KEY DEFINE MODE?	
4244	ODCF	CA	83 25	JZ DSPCHR ;NO - USE NORMAL ROUTINE	
4245	ODD2	CD	D7 13	CALL DCXB2D ;INPUT FROM KEYBOARD?	
4246	ODD5	C4	96 0D	CNZ SFKYOF ;NO - SWAP DISPLAY	
4247	ODD8	C2	83 25	JNZ DSPCHR ;AND USE NORMAL ROUTINE	
4248	ODDB	C3	FA 15	JMP FDESC1 ;YES - DISPLAY CHARACTER	
4249	ODDE	.	.	AND KILL "CURADV" FLAG	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 125
=====
4251      ODDE      . . .      ;*****
4252      ODDE      . . .      ; SFTRST - SOFT RESET *
4253      ODDE      . . .      ;*****
4254      ODDE      . . .      SFTRST EQU $
4255      ODDE      CD C0 16    CALL IOBSYC      ;WAIT UNTIL CTU'S FREE
4256      ODE1      F3 . .      DI                ;DISABLE INTERRUPTS
4257      ODE2      3E 01 .      MVI A,1          ;SET RESET TIMER FOR ONE
4258      ODE4      32 D0 FF    STA RSTTMR       ;SECOND ONLY
4259      ODE7      C3 19 01    JMP GO1          ;DO SOFT RESET
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 126
=====
4261      0DEA      . . .      ;*****
4262      0DEA      . . .      ; SO - SHIFT OUT *
4263      0DEA      . . .      ;*****
4264      0DEA      . . .      SHFTOT EQU $
4265      0DEA      CD E5 1A      CALL CHKSFK      ;DEFINE SOFT KEY MODE?
4266      0DED      C0 . .      RNZ              ;YES - DON'T SWITCH CHAR SET
4267      0DEE      3A 72 FF      LDA CHRSET      ;GET CURRENT ALT CHAR SET
4268      0DF1      . . .      SHFT1 EQU $
4269      0DF1      47 . .      MOV B,A         ;PUT NEW CHAR SET IN B-REG
4270      0DF2      3E 0B .      MVI A,SWCHAR    ;SET CHARACTER SWITCH IN
4271      0DF4      CD 08 48      CALL ZKBCTL     ;KEYBOARD FOR POSSIBLE
4272      0DF7      . . .      ;              FOREIGN MODE ENABLE
4273      0DF7      78 . .      MOV A,B         ;RECALL NEW CHARACTER SET
4274      0DF8      . . .      SHFT2 EQU $     ;ENTRY FOR SELF-TEST
4275      0DF8      06 0F .      MVI B,170       ;SET MASK TO SAVE DISPLAY
4276      0DFA      . . .      ;              ENHANCEMENT BITS
4277      0DFA      C3 43 24      JMP DISPC1      ;ADD CODE TO DISPLAY
4278      0DFD      . . .      ;*****
4279      0DFD      . . .      ; SI - SHIFT IN *
4280      0DFD      . . .      ;*****
4281      0DFD      . . .      SHFTIN EQU $
4282      0DFD      CD E5 1A      CALL CHKSFK     ;DEFINE SOFT KEY MODE?
4283      0E00      C0 . .      RNZ              ;YES - DON'T SWITCH CHAR SET
4284      0E01      AF . .      XRA A           ;SET FOR BASE CHARACTER
4285      0E02      C3 F1 0D      JMP SHFT1       ;SET CODE
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 127
=====
4287      0E05      . . .      ;*****
4288      0E05      . . .      ; STATUS - RETURN TERMINAL STATUS *
4289      0E05      . . .      ;*****
4290      0E05      . . .      STATUS EQU $
4291      0E05      01 00 02      LXI B,SSTAT ;SET BLOCK TRANSFER FOR
4292      0E08      C3 25 18      JMP SBLXFO ;FOR TERMINAL STATUS
4293      0E0B      . . .      ;*****
4294      0E0B      . . .      ; STATGO - TRANSMIT TERMINAL STATUS *
4295      0E0B      . . .      ;*****
4296      0E0B      . . .      STATGO EQU $
4297      0E0B      01 FF FD      LXI B,-1-SSTAT
4298      0E0E      CD 9B 11      CALL CLBLXF ;CLEAR STATUS PENDING FLAG
4299      0E11      06 5C .      MVI B,ABCKSL ;SEND <ESC>-<\>
4300      0E13      CD 1C 19      CALL ESCOUT
4301      0E16      21 22 19      LXI H,XPUTDC ;SET OUTPUT ROUTINE ADDRESS
4302      0E19      CD 26 0E      CALL STAPAR ;OUTPUT STATUS BITS
4303      0E1C      21 F7 FF      LXI H,ERRFLG ;CLEAR DATA COMM ERROR FLAG
4304      0E1F      7E . .      MOV A,M
4305      0E20      E6 FE .      ANI 377G-DCMERR
4306      0E22      77 . .      MOV M,A
4307      0E23      C3 4E 13      JMP SDTERM ;SEND TERMINATOR AND RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 128
4309	0E26	.	.	. ;*****	
4310	0E26	.	.	. ; STAPAR - OUTPUT STATUS BITS *	
4311	0E26	.	.	. ;*****	
4312	0E26	.	.	. ;	
4313	0E26	.	.	. ; ENTRY: H,L = ADDRESS OF OUTPUT ROUTINE	
4314	0E26	.	.	. ;	
4315	0E26	.	.	. ; EXIT : CNTFAD = ADDRESS OF OUTPUT ROUTINE	
4316	0E26	.	.	. ; ALL REGISTERS DESTROYED	
4317	0E26	.	.	. ;	
4318	0E26	.	.	. STAPAR EQU \$	
4319	0E26	22	CE	FF SHLD CNTFAD ;SET OUTPUT ROUTINE VECTOR	
4320	0E29	.	.	. ;	
4321	0E29	.	.	. ; OUTPUT SIZE OF RAM	
4322	0E29	.	.	. ;	
4323	0E29	3A	AB	FF LDA DSPBGN+1 ;COMPUTE NUMBER OF 256-BYTE	
4324	0E2C	2F	.	. CMA ;RAM BLOCKS IN DISPLAY	
4325	0E2D	3C	.	. INR A ;AREA	
4326	0E2E	CD	A6	0C CALL PARUT2 ;TRANSMIT MEMORY SIZE IN K'S	
4327	0E31	.	.	. ;	
4328	0E31	.	.	. ; OUTPUT KEYBOARD INTERFACE STRAP SETTINGS	
4329	0E31	.	.	. ;	
4330	0E31	3A	FB	FF LDA KBJMPR ;TRANSMIT STRAPS A-D	
4331	0E34	6F	.	. MOV L,A ;SAVE JUMPER VALUES	
4332	0E35	CD	A8	0C CALL PAROUT	
4333	0E38	7D	.	. MOV A,L ;RECALL JUMPER VALUES	
4334	0E39	CD	A4	0C CALL PARUT4 ;TRANSMIT STRAPS E-H	
4335	0E3C	.	.	. ;	
4336	0E3C	.	.	. ; OUTPUT LATCHING KEYS STATUS	
4337	0E3C	.	.	. ;	
4338	0E3C	3A	F3	FF LDA MDFLG2 ;GET TERMINAL MODE FLAGS 2	
4339	0E3F	E6	07	. ANI CAPSLK+BLKMDE+AUTGLF ;EXTRACT BITS	
4340	0E41	F6	08	. ORI 100 ;ADD BIT 3 TO INDICATE 2645	
4341	0E43	CD	A8	0C CALL PAROUT ;SEND LATCHING KEY STATUS	
4342	0E46	.	.	. ;	
4343	0E46	.	.	. ; OUTPUT TERMINAL (2640) TRANSFER PENDING FLAGS	
4344	0E46	.	.	. ;	
4345	0E46	2A	6F	FF LHLD MFLGS2 ;GET TERMINAL MODE FLAGS	
4346	0E49	7C	.	. MOV A,H ;MASK FOR SECONDARY STATUS	
4347	0E4A	E6	04	. ANI SSTAT2/256 ;PENDING BIT	
4348	0E4C	0F	.	. RRC ;SHIFT BIT INTO STATUS	
4349	0E4D	0F	.	. RRC ;RESPONSE POSITION	
4350	0E4E	0F	.	. RRC	
4351	0E4F	47	.	. MOV B,A	
4352	0E50	7C	.	. MOV A,H ;GET OTHER DISPLAY RELATED	
4353	0E51	E6	70	. ANI SENTER+SFCTKY+SCRSEN/256 ;XFR BITS	
4354	0E53	80	.	. ORA B ;ADD IN SECONDARY STATUS	
4355	0E54	CD	A4	0C CALL PAROT4 ;SEND TRANSFER PENDING BITS	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 129
4357	0E57	.	.	.	;
4358	0E57	.	.	.	;
4359	0E57	.	.	.	;
4360	0E57	06	00	.	MVI B,0 ;SET FOR NO I/O ERROR
4361	0E59	3A	4F	FF	LDA IOCERR ;GET I/O ERROR FLAG
4362	0E5C	FE	46	.	CPI F ;I/O ERROR OCCURRED?
4363	0E5E	C2	63	0E	JNZ STA010 ;NO - GET OTHER ERROR FLAGS
4364	0E61	06	20	.	MVI B,IOERRB ;YES - SET I/O ERROR BIT
4365	0E63	.	.	.	STA010 EQU \$
4366	0E63	3A	F7	FF	LDA ERRFLG ;GET THE ERROR FLAGS
4367	0E66	B0	.	.	ORA B ;MERGE WITH EXISTING BITS
4368	0E67	CD	A8	0C	CALL PAROUT ;TRANSMIT ERROR STATUS
4369	0E6A	.	.	.	;
4370	0E6A	.	.	.	;
4371	0E6A	.	.	.	;
4372	0E6A	7C	.	.	MOV A,H ;GET TERMINAL MODE 1 FLAGS
4373	0E6B	07	.	.	RLC ;PUT I/O DONE FLAG IN C-FLAG
4374	0E6C	7D	.	.	MOV A,L ;GET TERMINAL MODE 2 FLAGS
4375	0E6D	17	.	.	RAL ;ADD IN I/O DONE FLAG
4376	0E6E	47	.	.	MOV B,A ;SAVE TEMPORARY RESULTS
4377	0E6F	7C	.	.	MOV A,H ;RECALL TERMINAL MODE 1 FLAG
4378	0E70	0F	.	.	RRC ;PUT DEVICE STATUS INTO
4379	0E71	0F	.	.	RRC ;C-FLAG
4380	0E72	0F	.	.	RRC
4381	0E73	0F	.	.	RRC
4382	0E74	78	.	.	MOV A,B ;RECALL ACCUMULATED BITS
4383	0E75	17	.	.	RAL ;ADD IN DEVICE STATUS
4384	0E76	C3	A8	0C	JMP PAROUT ;SEND DEVICE XFR PENDING BIT

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 130
=====
4386      0E79      . . .      ;*****
4387      0E79      . . .      ; STCHR1 - SET INITIAL DISPLAY CHARACTER IN *
4388      0E79      . . .      ;   NEW DISPLAY BLOCK                               *
4389      0E79      . . .      ;*****
4390      0E79      . . .      ;
4391      0E79      . . .      ; ENTRY:  H,L = ADDRESS OF FIRST DISPLAY
4392      0E79      . . .      ;           IN BLOCK
4393      0E79      . . .      ;
4394      0E79      . . .      ; EXIT :  A = 0
4395      0E79      . . .      ;           H,L UNCHANGED
4396      0E79      . . .      ;
4397      0E79      . . .      STCHR1 EQU $
4398      0E79      3A F4 FF      LDA  MDFLG1      ;GET SOFT MODE FLAGS
4399      0E7C      E6 80 .      ANI  FORGN      ;FOREIGN MODE ENABLED?
4400      0E7E      3E CC .      MVI  A,EOL      ;(SET TO STORE EOL)
4401      0E80      CA 8B 0E      JZ   STC010     ;NO - STORE EOL ONLY
4402      0E83      2B . .      DCX  H          ;YES - STORE EOL AND DISPLAY
4403      0E84      77 . .      MOV  M,A        ;CONTROL BYTE TO CAUSE
4404      0E85      3A 29 48      LDA  FRSALT     ;FOREIGN CHARACTER SET TO
4405      0E88      F6 80 .      ORI  2000      ;BE DISPLAYED
4406      0E8A      23 . .      INX  H
4407      0E8B      . . .      STC010 EQU $
4408      0E8B      77 . .      MOV  M,A        ;STORE FIRST DISPLAY CHAR
4409      0E8C      AF . .      XRA  A          ;CLEAR A-REGISTER
4410      0E8D      C9 . .      RET            ;RETURN
=====
  
```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 131
4412	0E8E	. . .	;*****	
4413	0E8E	. . .	; TEST - PERFORM TERMINAL SELF TEST *	
4414	0E8E	. . .	;*****	
4415	0E8E	. . .	TEST EQU \$	
4416	0E8E	CD 78 11	CALL CKREDIT ;EDIT MODE ENABLED?	
4417	0E91	C0 . .	RNZ ;YES - DON'T DO SELF-TEST	
4418	0E92	3E 08 .	MVI A,CKIOKY	
4419	0E94	CD 08 48	CALL ZKBCTL ;I/O CONTROL KEY DOWN ALSO?	
4420	0E97	21 11 28	LXI H,TSTCTU ;(SET FOR CTU SELF-TEST)	
4421	0E9A	C2 E5 16	JNZ IORMGO ;YES - DO CTU SELF-TEST	
4422	0E9D	. . .	; NO - DO TERMINAL SELF-TEST	
4423	0E9D	. . .	; ;	
4424	0E9D	. . .	; PERFORM TERMINAL SELF-TEST	
4425	0E9D	. . .	; ;	
4426	0E9D	. . .	TRMTST EQU \$	
4427	0E9D	3A FA FF	LDA KBJMP2 ;GET KEYBOARD JUMPERS 2	
4428	0EA0	E6 04 .	ANI NOTEST ;SELF-TEST INHIBITED	
4429	0EA2	21 7B 10	LXI H,NOTSMS ;(SET MESSAGE ADDRESS)	
4430	0EA5	C2 30 1E	JNZ DSPMS1 ;YES - DISPLAY MSG AND EXIT	
4431	0EA8	3A 6E FF	LDA DFLGS ;GET DATA TRANSFER FLAGS	
4432	0EAB	E6 80 .	ANI XBF2DS ;DATA FROM I/O BUFFER	
4433	0EAD	C2 30 1E	JNZ DSPMS1 ;YES - DON'T DO SELF-TEST	
4434	0EB0	CD C0 16	CALL IOBSYC ;WAIT UNTIL CTU'S IDLE	
4435	0EB3	F3 . .	DI ;DISABLE INTERRUPTS	
4436	0EB4	3E 05 .	MVI A,STRTST ;SET KEYBOARD FOR SELF-TEST	
4437	0EB6	CD 08 48	CALL ZKBCTL ;START-UP	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 132
=====
4439      0EB9      . . .      ;*****
4440      0EB9      . . .      ; ROM TEST *
4441      0EB9      . . .      ; *
4442      0EB9      . . .      ; CALCULATE CHECKSUM *
4443      0EB9      . . .      ; FOR EACH 2K ROM *
4444      0EB9      . . .      ;*****
4445      0EB9      21 00 F8      LXI H,-NUM2K ;SET FOR START ADDRESS = 0
4446      0EBC      . . .      ;
4447      0EBC      . . .      TST010 EQU $
4448      0EBC      11 00 08      LXI D,NUM2K ;INCREMENT START ADDR BY 2K
4449      0EBF      19 . .      DAD D
4450      0EC0      . . .      ;
4451      0EC0      . . .      ; IS CURRENT ADDRESS A ROM?
4452      0EC0      . . .      ;
4453      0EC0      7C . .      MOV A,H ;PUT MSB INTO A-REGISTER
4454      0EC1      FE BF .      CPI 140000Q/256-1 ;ADDRESS > 48K?
4455      0EC3      D2 F9 0E      JNC TST050 ;YES - GO TO NEXT TEST
4456      0EC6      FE 80 .      CPI 100000Q/256 ;IN I/O SPACE?
4457      0EC8      CA BC 0E      JZ TST010 ;YES - GO TO NEXT ROM BLOCK
4458      0ECB      FE 88 .      CPI 104000Q/256
4459      0ECD      CA BC 0E      JZ TST010 ;YES - GO TO NEXT ROM BLOCK
4460      0ED0      CD F5 16      CALL IORMG1 ;DOES THE ROM EXIST?
4461      0ED3      CA DF 0E      JZ TST020 ;YES - CHECK THE ROM
4462      0ED6      AF . .      XRA A ;NO - CHECK FOR NO ROM
4463      0ED7      B5 . .      ORA L ;ROM INSTALLED?
4464      0ED8      CA BC 0E      JZ TST010 ;NO - GO TO NEXT ROM
4465      0EDB      7C . .      MOV A,H ;YES - REPORT POSSIBLE
4466      0EDC      C3 EA 0E      JMP TST030 ;MISPLACED ROM
4467      0EDF      . . .      ;*****
4468      0EDF      . . .      ; CALCULATE CHECKSUM *
4469      0EDF      . . .      ;*****
4470      0EDF      . . .      TST020 EQU $
4471      0EDF      2B . .      DCX H ;RESTORE START ADDRESS
4472      0EE0      16 08 .      MVI D,NUM2K/256 ;SET TO SUM 2K SPACE
4473      0EE2      CD 81 09      CALL CHKSUM ;CALCULATE CHECKSUM
4474      0EE5      3C . .      INR A ;= 377 ?
4475      0EE6      . . .      ;*****
4476      0EE6      CA BC 0E      JZ TST010 ;YES - DO NEXT ROM BLOCK
4477      0EE9      . . .      ;*****
4478      0EE9      AF . .      XRA A ;NO - REPORT BAD ROM
4479      0EEA      . . .      TST030 EQU $
4480      0EEA      11 61 10      LXI D,ROMERR ;SET ROM ERROR MESSAGE ADDR
4481      0EED      4F . .      MOV C,A ;SAVE EXPECTED VALUE
4482      0EEE      46 . .      MOV B,M ;GET VALUE FOUND
4483      0EEF      7C . .      MOV A,H ;CONVERT ROM ADDRESS TO
4484      0EF0      0F . .      RRC ;ROM NUMBER (0,2,4,...)
4485      0EF1      0F . .      RRC
4486      0EF2      6F . .      MOV L,A ;SET AS ERROR ADDRESS
4487      0EF3      26 00 .      MVI H,0
4488      0EF5      79 . .      MOV A,C ;RECALL EXPECTED VALUE
=====

```

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 133
=====
4489      0EF6      C3  1D  10          JMP  TST600      ;REPORT ERROR
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 134
4491	0EF9	. . .	;*****	
4492	0EF9	. . .	; RAM TEST	*
4493	0EF9	. . .	;	*
4494	0EF9	. . .	; CALCULATE CHECKSUM ON	*
4495	0EF9	. . .	; EACH 4K BLOCK.	*
4496	0EF9	. . .	; TEST EACH 256 BYTE SECTION	*
4497	0EF9	. . .	; RECHECK CHECKSUM.	*
4498	0EF9	. . .	;*****	
4499	0EF9	. . .	;	
4500	0EF9	. . .	; E = 0	
4501	0EF9	. . .	;	
4502	0EF9	. . .	TST050 EQU \$	
4503	0EF9	3E 80 .	MVI A,CRTOFF ;TURN OFF VIDEO	
4504	0EF8	32 20 87	STA IOCRRW	
4505	0EFE	21 00 FC	LXI H,IOBUF ;SET H,L TO I/O BUFFER #1	
4506	0F01	CD 30 12	CALL CLRAL1 ;CLEAR THE I/O BUFFER	
4507	0F04	44 . .	MOV B,H ;SET B,C = IOBUF2	
4508	0F05	4D . .	MOV C,L ;(H,L = IOBUF2)	
4509	0F06	16 10 .	MVI D,100000/256 ;SET D,E FOR 4K INCREMEN	
4510	0F08	63 . .	MOV H,E ;SET H TO 0 TO INDICATE STAR	
4511	0F09	02 . .	STAX B ;SET CHECKSUM FOR LAST	
4512	0F0A	. . .	; BLOCK TO ZERO	
4513	0F0A	. . .	;*****	
4514	0F0A	. . .	; CALCULATE CHECKSUM FOR EACH RAM BLOCK AND *	
4515	0F0A	. . .	; STORE CHECKSUM IN "IOBUF2" *	
4516	0F0A	. . .	;*****	
4517	0F0A	. . .	TST060 EQU \$	
4518	0F0A	CD 67 0C	CALL NXSBLK ;GET NEXT BLOCK ADDRESS	
4519	0F0D	CD 81 09	CALL CHKSUM ;COMPUTE CHECKSUM	
4520	0F10	02 . .	STAX B ;STORE CHECKSUM VALUE	
4521	0F11	C2 0A 0F	JNZ TST060 ;CONTINUE IF NOT LAST BLOCK	

13255

2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                PAGE 136
=====
4573     0F3B     C2  39  0F                JNZ  TST125      ;LOOP FOR 256 TIMES
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 137
4575	0F3E	.	.	; 6. CHECK VALUES. RESTORE ORIGINAL VALUE	
4576	0F3E	.	.	; B,C = IOBUF	
4577	0F3E	.	.	;	
4578	0F3E	.	.	TST130 EQU \$	
4579	0F3E	7D	.	MOV A,L	
4580	0F3F	AC	.	XRA H	
4581	0F40	2F	.	CMA	
4582	0F41	BE	.	CMP M ;SAME AS BEFORE?	
4583	0F42	C2	19 10	JNZ TST510 ;NO - REPORT ERROR WITH	
4584	0F45	.	.	EXPECTED/FOUND BYTES	
4585	0F45	0A	.	LDAX B	
4586	0F46	77	.	MOV M,A ;RESTORE	
4587	0F47	03	.	INX B	
4588	0F48	2C	.	INR L ;BLOCK COMPLETED?	
4589	0F49	C2	3E 0F	JNZ TST130 ;NO - DO NEXT BYTE	
4590	0F4C	.	.	*****	
4591	0F4C	.	.	; DONE WITH THIS SECTION. *	
4592	0F4C	.	.	; DO NEXT? *	
4593	0F4C	.	.	*****	
4594	0F4C	.	.	*****	
4595	0F4C	.	.	; SEE IF FIRST FAST RAM HAS BEEN TESTED	
4596	0F4C	3E	20 .	MVI A,400 ;FIRST FAST RAM BEING TESTED	
4597	0F4E	BB	.	CMP E	
4598	0F4F	C2	59 0F	JNZ TST135 ;NO	
4599	0F52	1E	00 .	MVI E,0 ;YES, TEST GRAPHICS RAM NOW	
4600	0F54	26	90 .	MVI H,FSTRAM/256-1	
4601	0F56	C3	18 0F	JMP TST090	
4602	0F59	.	.	TST135 EQU \$	
4603	0F59	.	.	*****	
4604	0F59	1C	.	INR E ;IF E = 0, WE JUST TESTED	
4605	0F5A	1D	.	DCR E ;FAST RAM	
4606	0F5B	C2	61 0F	JNZ TST140	
4607	0F5E	63	.	MOV H,E ;H=0, INDICATE START	
4608	0F5F	1E	80 .	MVI E,2000 ;BIT 7 = 1 MEANS 256	
4609	0F61	.	.	BYTE INCREMENTS	
4610	0F61	.	.	;	
4611	0F61	.	.	TST140 EQU \$	
4612	0F61	CD	67 0C	CALL NXSBLK ;GET NEXT BLOCK ADDRESS	
4613	0F64	B7	.	ORA A ;LAST BLOCK DONE?	
4614	0F65	C2	18 0F	JNZ TST090 ;NO, TEST NEXT	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 138
4616	0F68	.	.	*****	
4617	0F68	.	.	; CHECK ORIGINAL CHECKSUMS *	
4618	0F68	.	.	*****	
4619	0F68	.	.	;	
4620	0F68	.	.	; B,C = IOBUF2	
4621	0F68	.	.	;	
4622	0F68	59	.	MOV E,C ;SET E TO ZERO	
4623	0F69	26	FC	MVI H,IOBUF/256 ;SET H TO I/O BUFFER #1	
4624	0F6B	CD	30 12	CALL CLRAL1 ;CLEAR THE I/O BUFFER	
4625	0F6E	.	.	; (H,L) = IOBUF2, TOP HALF OF I/O BUFFER	
4626	0F6E	16	10	MVI D,100000/256 ;SET D,E FOR 4K INCREMEN	
4627	0F70	7E	.	MOV A,M ;GET CHECKSUM FOR TOP BLOCK	
4628	0F71	73	.	MOV M,E ;SET STORE BYTE TO ZERO	
4629	0F72	F5	.	PUSH PSW ;SAVE TOP BLOCK CHECKSUM	
4630	0F73	63	.	MOV H,E ;SET H TO 0 TO INDICATE STAR	
4631	0F74	.	.	*****	
4632	0F74	.	.	; RE-CALCULATE CHECKSUM FOR EACH RAM BLOCK AND *	
4633	0F74	.	.	; COMPARE TO INITIAL STORED VALUE *	
4634	0F74	.	.	*****	
4635	0F74	.	.	TST150 EQU \$	
4636	0F74	CD	67 0C	CALL NXSBLK ;GET NEXT BLOCK ADDRESS	
4637	0F77	CD	81 09	CALL CHKSUM ;COMPUTE CHECKSUM FOR BLOCK	
4638	0F7A	6F	.	MOV L,A ;SAVE COMPUTED VALUE IN L-RE	
4639	0F7B	CA	87 0F	JZ TST160 ;LAST BLOCK - CHECK 1ST VALU	
4640	0F7E	0A	.	LDAX B ;RECALL ORIGINAL CHECKSUM	
4641	0F7F	95	.	SUB L ;DO CHECKSUMS MATCH?	
4642	0F80	6F	.	MOV L,A ;(SET L TO ZERO IF TRUE)	
4643	0F81	CA	74 0F	JZ TST150 ;YES - GO TO NEXT BLOCK	
4644	0F84	C3	17 10	JMP TST500 ;NO - REPORT ERROR	
4645	0F87	.	.	;	
4646	0F87	.	.	TST160 EQU \$	
4647	0F87	F1	.	POP PSW ;RECALL 1ST STORED CHECKSUM	
4648	0F88	95	.	SUB L ;DO CHECKSUMS MATCH?	
4649	0F89	C2	17 10	JNZ TST500 ;NO - REPORT ERROR	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
4651	0F8C	.	.	. ;*****
4652	0F8C	.	.	. ; DO GRAPHICS TEST
4653	0F8C	CD	6D 60	CALL ZGTEST
4654	0F8F	.	.	. ;*****
4655	0F8F	.	.	. ;*****
4656	0F8F	.	.	. ; DISPLAY TEST PATTERN *
4657	0F8F	.	.	. ;*****
4658	0F8F	CD	14 48	CALL ZBELL ;SOUND THE BELL
4659	0F92	3E	C0 .	MVI A,3000 ;SET INITIAL CHARACTER SET
4660	0F94	.	.	. TST200 EQU \$
4661	0F94	D6	10 .	SUI 200 ;SET TO NEXT CHARACTER SET
4662	0F96	F5	.	. PUSH PSW ;SAVE CURRENT ENHANCEMENT
4663	0F97	AF	.	. XRA A ;SET CHARACTER TO NULL
4664	0F98	32	68 FF	STA TCHAR
4665	0F9B	.	.	. TST220 EQU \$
4666	0F9B	CD	66 23	CALL CRRET ;DO CR
4667	0F9E	CD	69 0B	CALL CONDLF ;DO LF IF WRAPAROUND DISABLED
4668	0FA1	F1	.	. POP PSW ;RECALL CURRENT ENHANCEMENT
4669	0FA2	F5	.	. PUSH PSW ;AND SAVE IT AGAIN
4670	0FA3	CD	F8 0D	CALL SHFT2 ;PUT ENHANCEMENT ON DISPLAY
4671	0FA6	.	.	. TST240 EQU \$
4672	0FA6	3A	68 FF	LDA TCHAR ;GET CURRENT ENHANCEMENT COD
4673	0FA9	32	89 FF	STA DCHAR ;STORE CHAR FOR DISPLAY
4674	0FAC	E6	07 .	ANI 7 ;EVERY 8 CHARS INSERT 2 BLNKS
4675	0FAE	FE	04 .	CPI 4 ;TIME TO ADD TWO BLANKS?
4676	0FB0	CC	86 21	CZ CURAD2 ;YES - ADVANCE CURSOR TWICE
4677	0FB3	CD	83 25	CALL DSPCHR ;DISPLAY THE CHARACTER
4678	0FB6	21	68 FF	LXI H,TCHAR
4679	0FB9	34	.	. INR M ;INCREMENT DISPLAY CHARACTER
4680	0FBA	7E	.	. MOV A,M ;GET NEW CHARACTER
4681	0FBB	FE	40 .	CPI 64
4682	0FBD	CA	9B 0F	JZ TST220 ;IF 64 THEN NEW LINE
4683	0FC0	B7	.	. ORA A ;ALL CHARACTERS DONE?
4684	0FC1	F2	A6 0F	JP TST240 ;NO - CONTINUE
4685	0FC4	CD	22 22	CALL CRLF ;YES - DOUBLE SPACE BETWEEN
4686	0FC7	F1	.	. POP PSW ;CHARACTER SETS
4687	0FC8	FE	80 .	CPI 2000 ;ALL CHARACTER SETS DONE?
4688	0FCA	C2	94 0F	JNZ TST200 ;NO - CONTINUE DISPLAY

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 140
4690	0FCD	. . .	;*****	
4691	0FCD	. . .	; DISPLAY ENHANCEMENT PATTERN *	
4692	0FCD	. . .	;*****	
4693	0FCD	F5 . .	PUSH PSW ;SAVE ENHANCEMENT CODE	
4694	0FCE	CD 69 0B	CALL CONDLF ;DO LF IF WRAPAROUND DISABLE	
4695	0FD1	. . .	TST420 EQU \$	
4696	0FD1	F1 . .	POP PSW ;RECALL CURRENT ENHANCEMENT	
4697	0FD2	F5 . .	PUSH PSW ;SAVE ENHANCEMENT AGAIN	
4698	0FD3	D6 40 .	SUI 100Q ;COMPUTE ASCII DISPLAY CODE	
4699	0FD5	CD 7D 25	CALL DSPTST ;DISPLAY THE CHARACTER	
4700	0FD8	F1 . .	POP PSW ;RECALL CURRENT ENHANCEMENT	
4701	0FD9	3C . .	INR A ;INCREMENT ENHANCEMENT	
4702	0FDA	FE 90 .	CPI 220Q ;LAST ENHANCEMENT DONE?	
4703	0FDC	CA E6 0F	JZ TST440 ;YES - DISPLAY STATUS	
4704	0FDF	F5 . .	PUSH PSW ;NO - SAVE ENHANCEMENT CODE	
4705	0FE0	CD 41 24	CALL DISPC0 ;ADD ENHANCEMENT TO DISPLAY	
4706	0FE3	C3 D1 0F	JMP TST420 ;DISPLAY ASCII DISPLAY CODE	
4707	0FE6	. . .	;	
4708	0FE6	. . .	TST440 EQU \$	
4709	0FE6	AF . .	XRA A	
4710	0FE7	CD 41 24	CALL DISPC0 ;RETURN TO NORMAL VIDEO	
4711	0FEA	CD 86 21	CALL CURAD2 ;ADVANCE CURSOR TWICE	
4712	0FED	. . .	;*****	
4713	0FED	. . .	; DISPLAY TERMINAL STATUS *	
4714	0FED	. . .	;*****	
4715	0FED	21 F7 FF	LXI H,ERRFLG ;SET ERROR FLAG TO	
4716	0FF0	7E . .	MOV A,M ;SELF-TEST SUCCESSFUL	
4717	0FF1	F6 02 .	ORI TESTOK	
4718	0FF3	77 . .	MOV M,A	
4719	0FF4	21 7D 25	LXI H,DSPTST ;SET H,L TO OUTPUT ROUTINE	
4720	0FF7	CD 26 0E	CALL STAPAR ;DISPLAY TERMINAL STATUS	
4721	0FFA	. . .	;*****	
4722	0FFA	. . .	; ROM BREAK 2	
4723	0FFA	C3 02 10	JMP ZBRK2C	
4724	0FFD	. . .	ORG ZBRK1+4000Q	
4725	1000	. . .	ZBRK2 EQU \$	
4726	1000	54 . .	DB VERSN ;ROM PRESENT FLAGS	
4727	1001	10 . .	DB ZBRK2/256	
4728	1002	. . .	ZBRK2C EQU \$	
4729	1002	. . .	;*****	
4730	1002	CD 89 21	CALL CURADV ;PUT SPACE BETWEEN STATUS	
4731	1005	CD C4 7D	CALL STA2G2	
4732	1008	CD 22 22	CALL CRLF	
4733	100B	CD 22 22	CALL CRLF	
4734	100E	. . .	;*****	
4735	100E	. . .	; TERMINATE SELF-TEST *	
4736	100E	. . .	;*****	
4737	100E	3E 06 .	MVI A,ENDTST ;RESTORE KEYBOARD LED'S	
4738	1010	CD 08 48	CALL ZKBCTL	
4739	1013	FB . .	EI ;RE-ENABLE INTERRUPTS	

13255

13255/90010

2648A MICROCODE LISTING 'PT91'

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                PAGE 141
=====
4740     1014     C3 AF 21          JMP CRADV1      ;RESET CURSOR ADVANCE FLAG
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 142
=====
4742     1017     . . .      TST500 EQU $           ;REPORT RAM ERROR
4743     1017     AF . .      XRA A                 ;SET Z TRUE FOR ADDRESS ONLY
4744     1018     6F . .      MOV L,A               ;FORCE L-REGISTER TO BE ZERO
4745     1019     . . .      TST510 EQU $
4746     1019     46 . .      MOV B,M               ;PUT VALUE FOUND INTO B-REG
4747     101A     11 65 10    LXI D,RAMERR         ;SET D,E TO ERROR MESSAGE
4748     101D     . . .      ;*****
4749     101D     . . .      ; REPORT ROM/RAM TEST ERROR *
4750     101D     . . .      ;*****
4751     101D     . . .      ;
4752     101D     . . .      ; ENTRY: D,E = ADDRESS AT WHICH ERROR OCCURRED
4753     101D     . . .      ; H,L = ERROR MESSAGE ADDRESS
4754     101D     . . .      ; Z - DISPLAY ERROR ADDRESS ONLY
4755     101D     . . .      ; NZ - DISPLAY PARAMETERS ALSO
4756     101D     . . .      ; A = EXPECTED VALUE
4757     101D     . . .      ; (H,L) = VALUE FOUND
4758     101D     . . .      ;
4759     101D     . . .      TST600 EQU $
4760     101D     EB . .      XCHG                  ;(H,L) = MESSAGE ADDRESS
4761     101E     . . .      ; (D,E) = ERROR ADDRESS
4762     101E     E5 . .      PUSH H                ;SAVE THE MESSAGE ADDRESS
4763     101F     21 7A 10    LXI H,ERREOP         ;SET EOP FOR SHORT MESSAGE
4764     1022     22 EB FF    SHLD MSGPT4
4765     1025     CA 38 10    JZ TST610             ;Z - SHOW ADDRESS ONLY
4766     1028     21 00 FD    LXI H,IOBUF2         ;SET BUFFER ADDRESS
4767     1028     22 EB FF    SHLD MSGPT4
4768     102E     C5 . .      PUSH B                ;SAVE VALUE FOUND
4769     102F     CD 02 09    CALL BINOC           ;CONVERT BINARY TO OCTAL
4770     1032     F1 . .      POP PSW               ;RECALL VALUE FOUND
4771     1033     CD 02 09    CALL BINOC           ;CONVERT BINARY TO OCTAL
4772     1036     36 CE .      MVI M,EOP            ;TERMINATE WITH "EOP"
4773     1038     . . .      TST610 EQU $
4774     1038     21 10 FD    LXI H,IOBUF2+16     ;CONVERT FAILURE ADDRESS
4775     1038     22 ED FF    SHLD MSGPT3
4776     103E     CD 2E 09    CALL BN2DEC          ;CONVERT TO DECIMAL ASCII
4777     1041     21 6C 10    LXI H,RXMERR         ;SET REST OF LITERAL
4778     1044     22 EF FF    SHLD MSGPT2
4779     1047     3A 6E FF    LDA DFLGS            ;GET DATA TRANSFER FLAGS
4780     104A     E6 01 .      ANI SDACOM           ;TEST FROM DATA COMM?
4781     104C     E1 . .      POP H                 ;(RECALL MESSAGE ADDRESS)
4782     104D     CA 85 13    JZ HANGUO            ;NO - SHOW MESSAGE AND HANG
4783     1050     C7 . .      RST 0                ;YES - RESET THE TERMINAL
=====

```

=====					PAGE 143	
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	
=====						
4785	1051	.	.	.	*****	
4786	1051	.	.	.	; MESSAGE STORAGE *	
4787	1051	.	.	.	*****	
4788	1051	.	.	.	BUFMSG EQU \$	
4789	1051	42	55	46	DB 'BUFFER OVERFLOW',EOP	
4790	1061	.	.	.	;	
4791	1061	.	.	.	ROMERR EQU \$	
4792	1061	52	4F	4D	DB 'ROM',0	
4793	1065	.	.	.	;	
4794	1065	.	.	.	RAMERR EQU \$	
4795	1065	52	41	4D	DB 'RAM',0	
4796	1069	.	.	.	;	
4797	1069	.	.	.	INERMS EQU \$	
4798	1069	49	2F	4F	DB 'I/O'	
4799	106C	.	.	.	;	
4800	106C	.	.	.	RXMERR EQU \$	
4801	106C	20	45	52	DB ' ERROR ',0	
4802	1074	.	.	.	;	
4803	1074	.	.	.	LDRMSG EQU \$	
4804	1074	4C	4F	41	DB 'LOADER'	
4805	107A	.	.	.	ERREOP EQU \$	
4806	107A	CE	.	.	DB EOP	
4807	107B	.	.	.	;	
4808	107B	.	.	.	NOTSMS EQU \$	
4809	107B	4E	4F	20	DB 'NO TEST',EOP	
4810	1083	.	.	.	;	
4811	1083	.	.	.	NODRV R EQU \$	
4812	1083	4E	4F	20	DB 'NO DEVICE DRIVER',EOP	
4813	1094	.	.	.	;	
4814	1094	.	.	.	TRMRDY EQU \$	
4815	1094	54	45	52	DB 'TERMINAL READY',EOP	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 144
4817	10A3	.	.	*****	
4818	10A3	.	.	; TOPUPD - UPDATE TOP LINE POINTERS *	
4819	10A3	.	.	*****	
4820	10A3	.	.	TOPUPD EQU \$	
4821	10A3	23	.	INX H ;PUT THE MSB INTO THE	
4822	10A4	46	.	MOV B,M ;B-REGISTER	
4823	10A5	4F	.	MOV C,A ;SAVE TOP LINE'S LSB IN C-RE	
4824	10A6	21	A3	FF LXI H,TLINO ;UPDATE TOP LINE NUMBER	
4825	10A9	7A	.	MOV A,D	
4826	10AA	87	.	ORA A ;IS TLINO TO BE RESET?	
4827	10AB	CA	AF	10 JZ TOP100 ;YES	
4828	10AE	86	.	ADD M ;NO - INCREMENT OR DECREMENT	
4829	10AF	.	.	TOP100 EQU \$	
4830	10AF	77	.	MOV M,A ;STORE UPDATED TLINO	
4831	10B0	.	.	TOPUP1 EQU \$	
4832	10B0	60	.	MOV H,H ;SET NEW TOP LINE POINTER	
4833	10B1	69	.	MOV L,C	
4834	10B2	22	CB	FF SHLD TUPLIN	
4835	10B5	3A	F8	FF LDA CMFLGS ;GET COMMON FLAGS	
4836	10B8	E6	08	. ANI DEFSKY ;SOFT KEY DEFINE MODE?	
4837	10BA	C0	.	. RNZ ;YES - DON'T CHANGE SCREEN	
4838	10BB	.	.	. ;!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!*	
4839	10BB	CD	23	60 CALL ZMUCHK ;AUTO PLOT MENU ON?	
4840	10BE	C0	.	. RNZ ;YES, DONT CHANGE	
4841	10BF	21	B2	90 LXI H,ZGFLG1 ;A/N DISPLAY INHIBITED?	
4842	10C2	3E	20	. MVI A,AVINHB	
4843	10C4	A6	.	. ANA M	
4844	10C5	C0	.	. RNZ ;YES, DONT CHANGE	
4845	10C6	.	.	. ;*****	
4846	10C6	21	FF	FF LXI H,DISPST+1 ;GET DISPLAY START ADDRESS	
4847	10C9	0B	.	. DCX B ;SET TO FIRST CHAR ADDRESS	
4848	10CA	.	.	. ;*****	
4849	10CA	.	.	. ; DISLNK - STORE LINK IN DISPLAY AREA *	
4850	10CA	.	.	. ;*****	
4851	10CA	.	.	. ;	
4852	10CA	.	.	. ; ENTRY: B,C = LINK TO BE STORED	
4853	10CA	.	.	. ; H,L = STORE ADDRESS FOR MS6 PART	
4854	10CA	.	.	. ;	
4855	10CA	.	.	. ; EXIT : H,L = LSB OF STORE ADDRESS	
4856	10CA	.	.	. ; A DESTROYED	
4857	10CA	.	.	. ; INTERRUPTS ENABLED	
4858	10CA	.	.	. ;	
4859	10CA	.	.	. DISLNK EQU \$	
4860	10CA	3E	60	. MVI A,DMAOFF ;SET TO TURN OFF THE DMA	
4861	10CC	F3	.	. OI ;DISABLE INTERRUPTS	
4862	10CD	.	.	. ;!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!*	
4863	10CD	CD	0B	60 CALL ZANCHK ;TURN OFF DMA	
4864	10D0	.	.	. ;*****	
4865	10D0	70	.	. MOV M,B ;STORE LINK'S MSB	
4866	10D1	2B	.	. DCX H	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 145
=====
4867     10D2     71 . .      MOV M,C          ;STORE LINK'S MSB
4868     10D3     . . .      DISLN1 EQU $     ;SET CURSOR ROW POSITION
4869     10D3     3A C0 FF    LDA CURROW      ;TURN DMA BACK ON WITH
4870     10D6     . . .      DISLN2 EQU $
4871     10D6     . . .      ;!!!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!*
4872     10D6     CD 0B 60    CALL ZANCHK     ;CURRENT CURSOR ROW ADDRESS
4873     10D9     . . .      ;*****
4874     10D9     . . .      DISLN3 EQU $
4875     10D9     FB . .      EI              ;RE-ENABLE INTERRUPTS
4876     10DA     . . .      DISLN4 EQU $     ;RE-ENABLE RESET KEY
4877     10DA     3E 02 .      MVI A,RSTON
4878     10DC     32 80 83    STA IOKBCO
4879     10DF     C9 . .      RET             ;RETURN
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 146
=====
4881     10E0      . . .      ;*****
4882     10E0      . . .      ; TYPSET - SET TYPE DEFINITION *
4883     10E0      . . .      ;*****
4884     10E0      . . .      TYPSET EQU $
4885     10E0      CD CF 1A    CALL CHKFMS      ;FORMAT/SOFT KEY DEFINE MODE
4886     10E3      C0 . .     RNZ              ;YES - DO SET TYPE
4887     10E4      3A 89 FF    LDA DCHAR       ;NO - COMPUTE TYPE DEFINITIO
4888     10E7      C6 8F .     ADI ALPHA-ZERO-6 ;CHARACTER
4889     10E9      C3 43 24    JMP DISPC1      ;ADD CHARACTER TO DISPLAY
4890     10EC      . . .      ;*****
4891     10EC      . . .      ; SFKCHK - SOFT KEY ATTRIBUTE CHECK *
4892     10EC      . . .      ;*****
4893     10EC      . . .      SFKCHK EQU $
4894     10EC      E6 DF .     ANI 377Q-40Q    ;FORCE INPUT TO UPPER CASE
4895     10EE      2A C3 FF    LHLD CURADR     ;RECALL CHARACTER ADDRESS
4896     10F1      77 . .     MOV M,A         ;STORE UPPER CASE VALUE
4897     10F2      FE 4E .     CPI N           ;NORMAL ATTRIBUTE SET?
4898     10F4      C8 . .     RZ             ;YES - RETURN SUCCESSFUL
4899     10F5      FE 4C .     CPI L           ;LOCAL ATTRIBUTE SET?
4900     10F7      C8 . .     RZ             ;YES - RETURN SUCCESSFUL
4901     10F8      FE 54 .     CPI T           ;TRANSMIT ONLY SET?
4902     10FA      C8 . .     RZ             ;YES - RETURN SUCCESSFUL
4903     10FB      70 . .     MOV M,B        ;NO - RESTORE ORIGINAL
4904     10FC      C9 . .     RET            ;ATTRIBUTE AND RETURN NZ
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 147
4906	10FD	.	.	;	
4907	10FD	.	.	; * * * * *	
4908	10FD	.	.	;	
4909	10FD	.	.	; XMS2DS - TRANSFER MESSAGE TO NORMAL DISPLAY	
4910	10FD	.	.	;	
4911	10FD	.	.	; ENTRY: H,L = POINTER TO MESSAGE	
4912	10FD	.	.	;	
4913	10FD	.	.	; EXIT : A-L DESTROYED	
4914	10FD	.	.	; Z - TERMINATED BY A NULL BYTE	
4915	10FD	.	.	; NZ - TERMINATED BY AN EOP	
4916	10FD	.	.	;	
4917	10FD	.	.	XMD000 EQU \$	
4918	10FD	CD	7D 25	CALL DSPTST ;DISPLAY ASCII CHARACTER AND	
4919	1100	.	.	ADVANCE CURSOR	
4920	1100	.	.	XMD010 EQU \$	
4921	1100	E1	.	POP H ;RESTORE H AND L	
4922	1101	23	.	INX H ;MOVE TO NEXT BYTE	
4923	1102	.	.	; PROCESS THE NEXT BYTE	
4924	1102	.	.	XMS2DS EQU \$	
4925	1102	7E	.	MOV A,M ;SET THE SOURCE BYTE	
4926	1103	B7	.	ORA A ;IS IT A NULL BYTE?	
4927	1104	C8	.	RZ ;YES - RETURN (Z - TRUE)	
4928	1105	FE	CE	CPI EOP ;IS IT END OF PAGE FLAG?	
4929	1107	CA	22 11	JZ XMD030 ;YES - EXIT	
4930	110A	E5	.	PUSH H ;NO - SAVE H,L	
4931	110B	FE	CC	CPI EOL ;IS IT AN END OF LINE?	
4932	110D	CA	1C 11	JZ XMD020 ;YES - START A NEW LINE	
4933	1110	B7	.	ORA A ;IS CHARACTER ASCII?	
4934	1111	F2	FD 10	JP XMD000 ;YES - DISPLAY IT	
4935	1114	06	00	MVI B,0 ;NO - FORCE ENHANCEMENT CODE	
4936	1116	CD	45 24	CALL DISPC2 ;TO BE STORED AS IS	
4937	1119	C3	00 11	JMP XMD010 ;GO TO NEXT BYTE	
4938	111C	.	.	;	
4939	111C	.	.	; EOL CODE - TERMINATE THE LINE	
4940	111C	.	.	;	
4941	111C	.	.	XMD020 EQU \$	
4942	111C	CD	22 22	CALL CRLF ;PERFORM RETURN AND LINE FEE	
4943	111F	C3	00 11	JMP XMD010 ;DO NEXT BYTE	
4944	1122	.	.	;	
4945	1122	.	.	; EOP CODE - TERMINATE LINE AND EXIT	
4946	1122	.	.	;	
4947	1122	.	.	XMD030 EQU \$	
4948	1122	CD	22 22	CALL CRLF ;PUT CURSOR IN NEXT LINE	
4949	1125	B4	.	ORA H ;SET Z FALSE	
4950	1126	C9	.	RET ;RETURN TERMINATED BY EOP	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 148
=====
4952     1127      . . .      ;*****
4953     1127      . . .      ; CARRET - PERFORM DISPLAY FUNCTIONS RETURN *
4954     1127      . . .      ;*****
4955     1127      . . .      CARRET EQU $
4956     1127      CD E5 1A    CALL CHKSFK      ;SOFT KEY DEFINE MODE?
4957     112A      CA 36 11    JZ CAR010        ;NO - DJ NORMAL PROCESSING
4958     112D      CD D7 13    CALL DCXB2D      ;DATA FROM KEYBOARD?
4959     1130      CA 83 25    JZ DSPCHR        ;YES - DISPLAY RETURN CODE
4960     1133      CD 96 0D    CALL SFKYOF      ;NO - RESTORE NORMAL DISPLAY
4961     1136      . . .      CAR010 EQU $
4962     1136      21 22 22    LXI H,CRLF      ;SET NORMAL ROUTINE EXIT
4963     1139      C3 8F 25    JMP DSPCH0       ;DISPLAY THE CHARACTER
=====
  
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 149
4965	113C	. . .	;*****	
4966	113C	. . .	; CHKLIM - CHECK PARAMETER BOUNDARY CONDITIONS *	
4967	113C	. . .	;*****	
4968	113C	. . .	;	
4969	113C	. . .	; ENTRY: B = CURRENT VALUE	
4970	113C	. . .	; C = MAXIMUM ALLOWABLE VALUE	
4971	113C	. . .	; D,E = ADDRESS OF PARAMETER TO BE SET	
4972	113C	. . .	; IODATA = PARAMETER VALUE (2 BYTES)	
4973	113C	. . .	; IOPSGN = -1, NEGATIVE ADJUSTMENT	
4974	113C	. . .	; = 0, ABSOLUTE VALUE	
4975	113C	. . .	; = +1, POSITIVE ADJUSTMENT	
4976	113C	. . .	;	
4977	113C	. . .	; EXIT : NEW VALUE IN WORD ADDRESSED BY D,E	
4978	113C	. . .	; A,C,H,L DESTROYED	
4979	113C	. . .	;	
4980	113C	. . .	; THIS ROUTINE SET THE NEW VALUE BY EITHER	
4981	113C	. . .	; OR ABSOLUTE ADJUST WITHING THE LIMITS OF	
4982	113C	. . .	; ZERO AND THE MAXIMUM ALLOWABLE AS SPECIFIED	
4983	113C	. . .	; THE C-REGISTER ON ENTRY	
4984	113C	. . .	;	
4985	113C	. . .	; THE LARGEST MAXIMUM VALUE IS 255	
4986	113C	. . .	;	
4987	113C	. . .	CHKLIO EQU \$	
4988	113C	3A DD FF	LDA IOCSGN ;SET PARAMETER SIGN TO	
4989	113F	32 DC FF	STA IOPSGN ;INPUT SIGN	
4990	1142	. . .	CHKLIM EQU \$	
4991	1142	3A DF FF	LDA IODATA+1 ;GET MSB OF INPUT VALUE	
4992	1145	B7 . .	ORA A ;MAXIMUM EXCEEDED?	
4993	1146	3A DC FF	LDA IOPSGN ;(GET PARAMETER SIGN)	
4994	1149	CA 53 11	JZ CHK050 ;NO - CONTINUE EVALUATION	
4995	114C	87 . .	ADD A ;NEGATIVE ADJUSTMENT?	
4996	114D	F2 62 11	JP CHK070 ;NO - SET TO MAXIMUM VALUE	
4997	1150	. . .	;	
4998	1150	. . .	; DEFAULT TO MINIMUM VALUE (0)	
4999	1150	. . .	;	
5000	1150	. . .	CHK010 EQU \$;SET TO ZERU	
5001	1150	AF . .	XRA A	
5002	1151	12 . .	STAX D ;STORE NEW VALUE	
5003	1152	C9 . .	RET ;RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 150
5005	1153	.	.	;	
5006	1153	.	.	; PARAMETER < 256, EVALUATE FOR RELATIVE AMOUNT	
5007	1153	.	.	;	
5008	1153	.	.	CHK050 EQU \$	
5009	1153	21	DE FF	LXI H,IODATA ;SET H,L TO GET INPUT VALUE	
5010	1156	87	.	ADD A ;RELATIVE POSITIONING?	
5011	1157	78	.	MOV A,B ;(LOAD CURRENT VALUE)	
5012	1158	FA	6C 11	JM CHK160 ;MINUS - SUBTRACT INPUT	
5013	1158	C2	65 11	JNZ CHK150 ;PLUS - ADD INPUT	
5014	115E	7E	.	MOV A,M ;NONE - ABSOLUTE ASSIGNMENT	
5015	115F	.	.	;	
5016	115F	.	.	; CHECK UPPER LIMIT + 1	
5017	115F	.	.	;	
5018	115F	.	.	CHK060 EQU \$	
5019	115F	12	.	STAX D ;STORE ASSIGNED VALUE	
5020	1160	B9	.	CMP C ;MAXIMUM EXCEEDED?	
5021	1161	D8	.	RC ;NO - RETURN	
5022	1162	.	.	CHK070 EQU \$;YES - USE MAXIMUM VALUE	
5023	1162	79	.	MOV A,C	
5024	1163	.	.	*****	
5025	1163	.	.	; STORE PARAMETER VALUE *	
5026	1163	.	.	*****	
5027	1163	.	.	CHK100 EQU \$	
5028	1163	12	.	STAX D ;STORE PARAMETER VALUE	
5029	1164	C9	.	RET ;RETURN	
5030	1165	.	.	;	
5031	1165	.	.	; POSITIVE ADJUSTMENT - ADD INPUT	
5032	1165	.	.	;	
5033	1165	.	.	CHK150 EQU \$	
5034	1165	86	.	ADD M ;OVERFLOW?	
5035	1166	D2	5F 11	JNC CHK060 ;NO - USE SPECIFIED VALUE	
5036	1169	C3	62 11	JMP CHK070 ;YES - USE MAXIMUM VALUE	
5037	116C	.	.	;	
5038	116C	.	.	; NEGATIVE ADJUSTMENT - SUBTRACT INPUT	
5039	116C	.	.	;	
5040	116C	.	.	CHK160 EQU \$	
5041	116C	96	.	SUB M ;UNDERFLOW?	
5042	116D	DA	50 11	JC CHK010 ;YES - USE ZERO	
5043	1170	12	.	STAX D ;NO - USE COMPUTED VALUE	
5044	1171	C9	.	RET ;RETURN	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 151
5046	1172	. . .	;*****	
5047	1172	. . .	; CKDSPF - CHECK FOR DISPLAY FUNCTIONS ENABLED *	
5048	1172	. . .	;*****	
5049	1172	. . .	CKDSPF EQU \$	
5050	1172	3A F4 FF	LDA MDFLG1 ;GET SOFT MODE FLAGS	
5051	1175	E6 01 .	ANI DSPFNC ;MASK FUR DISPLAY FUNCTIONS	
5052	1177	C9 . .	RET ;FLAG AND RETURN	
5053	1178	. . .	;*****	
5054	1178	. . .	; CKREDIT - CHECK FOR EDIT MODE ENABLED *	
5055	1178	. . .	;*****	
5056	1178	. . .	CKREDIT EQU \$	
5057	1178	3A F4 FF	LDA MDFLG1 ;GET SOFT MODE FLAGS	
5058	1178	E6 10 .	ANI EDIT ;MASK FUR EDIT FLAG AND	
5059	117D	C9 . .	RET ;RETURN	
5060	117E	. . .	;*****	
5061	117E	. . .	; GTMODE - DETERMINE MODE OF TERMINAL *	
5062	117E	. . .	; Z = TRUE IF CHARACTER MODE *	
5063	117E	. . .	; Z = FALSE IF PAGE MODE *	
5064	117E	. . .	;*****	
5065	117E	. . .	GTMOD1 EQU \$	
5066	117E	3A 64 FF	LDA IOFLG2 ;GET I/O FLAGS	
5067	1181	E6 20 .	ANI XDS2BF ;DISPLAY TO BUFFER TRANSFER?	
5068	1183	C0 . .	RNZ ;YES - RETURN PAGE MODE	
5069	1184	. . .	GTMODE EQU \$;NO - CHECK REAL PAGE MODE	
5070	1184	3A F3 FF	LDA MDFLG2 ;GET TERMINAL MODE FLAGS 2	
5071	1187	E6 02 .	ANI BLKMDE ;BLUCK MODE ENABLED?	
5072	1189	C8 . .	RZ ;NO - RETURN (Z=TRUE)	
5073	118A	. . .	;	
5074	118A	. . .	; CKLNMD - CHECK LINE MODE	
5075	118A	. . .	;	
5076	118A	. . .	; EXIT : Z = TRUE, LINE MODE	
5077	118A	. . .	; = FALSE, PAGE MODE	
5078	118A	. . .	; A,L DESTROYED	
5079	118A	. . .	;	
5080	118A	. . .	CKLNMD EQU \$	
5081	118A	3A F8 FF	LDA KBJMPR ;GET THE STRAP SETTINGS	
5082	118D	E6 08 .	ANI PAGSTR ;SET Z-FLAG	
5083	118F	C9 . .	RET ;RETURN	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 152
=====
5085     1190      . . .      ;*****
5086     1190      . . .      ; CKPROT - CHECK PROTECT STATUS OF CURRENT *
5087     1190      . . .      ;   CURSOR LOCATION                               *
5088     1190      . . .      ;*****
5089     1190      . . .      CKPROT EQU $
5090     1190      3A C2 FF    LDA  PROFLO      ;GET PROTECT FLAG
5091     1193      3C . .     INR  A           ;SET Z-FLAG (-1 => PROTECTED
5092     1194      C9 . .     RET              ;RETURN
5093     1195      . . .      ;*****
5094     1195      . . .      ; CKRMTE - CHECK FOR REMOTE MODE ENABLED *
5095     1195      . . .      ;*****
5096     1195      . . .      CKRMTE EQU $
5097     1195      3A F8 FF    LDA  CMFLGS      ;GET COMMON FLAGS
5098     1198      E6 10 .     ANI  REMSET      ;MASK FOR REMOTE FLAG
5099     119A      C9 . .     RET              ;RET (NZ => YES; Z => NO)
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 153
5101	119B	.	.	*****	
5102	119B	.	.	; CLBLXF - CLEAR BLOCK TRANSFER PENDING FLAG *	
5103	119B	.	.	*****	
5104	119B	.	.	;	
5105	119B	.	.	; ENTRY: B = 377B-(FLAG TO CLEAR FROM MFLGS)	
5106	119B	.	.	; C = 377B-(FLAG TO CLEAR FROM MFLGS2)	
5107	119B	.	.	;	
5108	119B	.	.	; EXIT : H = BASEH	
5109	119B	.	.	; A,B,L DESTROYED	
5110	119B	.	.	;	
5111	119B	.	.	; CLEARS THE SPECIFIED TRANSFER PENDING FLAG	
5112	119B	.	.	; FROM "MFLGS". IF NO OTHER TRANSFER IS PENDING,	
5113	119B	.	.	; THEN THE KEYBOARD IS UNLOCKED. OTHERWISE,	
5114	119B	.	.	; THE NEXT TRANSFER PENDING IS SET UP.	
5115	119B	.	.	;	
5116	119B	.	.	CLBLXF EQU \$	
5117	119B	2A	6F FF	LHLD MFLGS2 ;GET TRANSFER PENDING FLAGS	
5118	119E	78	.	MOV A,B	
5119	119F	A4	.	ANA H ;CLEAR FLAG FROM "MFLGS"	
5120	11A0	67	.	MOV H,A	
5121	11A1	79	.	MOV A,C	
5122	11A2	A5	.	ANA L ;CLEAR FLAG FROM "MFLGS2"	
5123	11A3	6F	.	MOV L,A	
5124	11A4	22	6F FF	SHLD MFLGS2 ;STORE NEW FLAG VALUES	
5125	11A7	E6	03 .	ANI SBINRY+SDVREC	
5126	11A9	B4	.	ORA H ;ANY MORE TRANSFER PENDING?	
5127	11AA	.	.	*****	
5128	11AA	E5	.	PUSH H ;SAVE HL	
5129	11AB	21	68 90	LXI H,ZGSBLK ;ANY GRAPHICS PENDING?	
5130	11AE	B6	.	ORA M	
5131	11AF	E1	.	POP H ;RESTORE HL	
5132	11B0	.	.	*****	
5133	11B0	01	00 00	LXI B,0 ;(SET FOR NULL FLAGS SET)	
5134	11B3	C2	25 18	JNZ SBLXF0 ;YES - SET UP NEXT BLOCK XFR	
5135	11B6	CD	46 17	CALL KBEN ;NO - RE-ENABLE KEYBOARD	
5136	11B9	.	.	;	
5137	11B9	.	.	; CLRXON - CLEAR BLOCK TRANSFER TRIGGER	
5138	11B9	.	.	;	
5139	11B9	.	.	CLRXON EQU \$	
5140	11B9	3E	00 .	MVI A,CLRTRG ;CLEAR BLOCK TRANSFER TRIGGE	
5141	11BB	CD	73 13	CALL DCMCTL ;PERFORM DATACOM CONTROL	
5142	11BE	37	.	STC ;SET C-FLAG TRUE AND	
5143	11BF	C9	.	RET ;RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 154
5145	11C0	.	.	*****	
5146	11C0	.	.	; CLEARS - CLEAR DISPLAY FROM CURSOR POSITION *	
5147	11C0	.	.	*****	
5148	11C0	.	.	CLEARS EQU \$;CLEAR UNPROTECTED FIELDS	
5149	11C0	3E	FE	MVI A,377Q-SDACOM ;ONLY BY CLEARING DATA	
5150	11C2	CD	53	17 CALL CLRDFL ;COMM INPUT FLAG	
5151	11C5	CD	CF	1A CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE	
5152	11C8	C2	FA	11 JNZ CLS100 ;YES - CLEAR FIELDS ONLY	
5153	11CB	CD	95	1D CALL CLEARL ;CLEAR LINE FROM CURSOR	
5154	11CE	F8	.	RM ;RETURN IF LINE NOT FOUND	
5155	11CF	2A	C9	FF LHLD LSTLIN ;GET CURRENT LINE ADDRESS	
5156	11D2	7E	.	MOV A,M ;GET THE LSB VALUE	
5157	11D3	B7	.	ORA A ;NEXT LINE EXIST (LSB # 0)?	
5158	11D4	C8	.	RZ ;NO - RETURN	
5159	11D5	E5	.	PUSH H ;YES - ADD SUCCEEDING LINES	
5160	11D6	36	00	. MVI M,0 ;TO FREE BLOCKS LIST	
5161	11D8	23	.	. INX H ;SET NEXT LINE POINTER TO	
5162	11D9	56	.	. MOV D,M ;INDICATE NO NEXT LINE	
5163	11DA	36	CE	. MVI M,EUP	
5164	11DC	5F	.	MOV E,A ;SET D,E TO TOP NEXT LINE	
5165	11DD	2A	AC	FF LHLD FRBLKS ;GET CURRENT FREE BLOCKS HEA	
5166	11E0	EB	.	. XCHG ;SET PREVIOUS LINE POINTER	
5167	11E1	23	.	. INX H ;IN FIRST SUCCEEDING LINE	
5168	11E2	23	.	. INX H ;TO CURRENT FREE BLOCKS	
5169	11E3	23	.	. INX H ;HEAD	
5170	11E4	73	.	. MOV M,E	
5171	11E5	23	.	. INX H	
5172	11E6	72	.	. MOV M,D	
5173	11E7	2A	A1	FF LHLD LLINE ;SET FREE BLOCKS HEAD TO	
5174	11EA	22	AC	FF SHLD FRBLKS ;CURRENT LAST LINE	
5175	11ED	E1	.	. POP H ;SET LAST LINE ADDRESS TO	
5176	11EE	22	A1	FF SHLD LLINE ;CURRENT LINE	
5177	11F1	.	.	. *****	
5178	11F1	.	.	. ; MEMORY RELEASED *	
5179	11F1	.	.	. ; CLEAR LOCK FLAGS *	
5180	11F1	.	.	. *****	
5181	11F1	.	.	. MLKOF EQU \$	
5182	11F1	3A	F4	FF LDA MDFLG1	
5183	11F4	E6	04	. ANI MEMLOK ;MEMORY LOCK ENABLED?	
5184	11F6	C8	.	. RZ ;NO - RETURN	
5185	11F7	C3	E2	0B JMP MLO010 ;YES - SET LED ON WO/BLINKIN	


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 155
=====
5187     11FA      .      .      .      ;*****
5188     11FA      .      .      .      ; FORMAT MODE CLEAR SCREEN FROM CURSOR *
5189     11FA      .      .      .      ;*****
5190     11FA      .      .      .      CLS100 EQU $
5191     11FA      F4     CA     07      CP      RCADR4      ;LOCATE CHAR IF FORMAT MODE
5192     11FD      F8      .      .      RM      ;RETURN IF NOT FOUND OR IN
5193     11FE      .      .      .      ;
5194     11FE      C2     05     12      JNZ     CLS110      ;PAST EOL - START AT NEXT FL
5195     1201      04      .      .      INR     B           ;CURSOR IN UNPROTECTED FIELD
5196     1202      C2     0E     12      JNZ     CLS130      ;YES - CLEAR REST OF FIELD
5197     1205      .      .      .      ;*****
5198     1205      .      .      .      ; CURSOR IN PROTECTED FIELD *
5199     1205      .      .      .      ; TAB TO NEXT UNPROTECTED FIELD *
5200     1205      .      .      .      ;*****
5201     1205      .      .      .      CLS110 EQU $
5202     1205      CD     3A     1F      CALL   FLDSR      ;SEARCH TO NEXT FIELD
5203     1208      C8      .      .      RZ      ;NO MORE FIELDS - RETURN
5204     1209      1A      .      .      LDAX   D          ;GET END PROTECT CHARACTER
5205     120A      .      .      .      CLS120 EQU $
5206     120A      32     C5     FF      STA     LSTFMT     ;SET LAST FORMAT CODE
5207     120D      1B      .      .      DCX   D           ;SKIP OVER "ENDPR" CHAR
5208     120E      .      .      .      ;*****
5209     120E      .      .      .      ; CLEAR UNPROTECTED FIELD *
5210     120E      .      .      .      ;*****
5211     120E      .      .      .      CLS130 EQU $
5212     120E      CD     F5     1D      CALL   CLER01     ;CLEAR FIELD
5213     1211      FE     CE     .      CPI     EOP        ;TERMINATION AT END OF PAGE?
5214     1213      C8      .      .      RZ      ;YES - RETURN
5215     1214      .      .      .      ;*****
5216     1214      .      .      .      ; SEARCH FOR NEXT UNPROTECTED FIELD *
5217     1214      .      .      .      ;*****
5218     1214      1B      .      .      DCX   D           ;ADJUST ADDRESS TO NEXT CHAR
5219     1215      .      .      .      CLS200 EQU $
5220     1215      13      .      .      INX   D           ;ADJUST ADDRESS TO PREV CHAR
5221     1216      .      .      .      CLS210 EQU $
5222     1216      CD     90     0C      CALL   NXTCHR     ;GET NEXT CHARACTER
5223     1219      C2     15     12      JNZ     CLS200     ;SKIP OVER EOL LINK
5224     121C      FE     C1     .      CPI     ENDPR      ;NEW FIELD?
5225     121E      CA     0A     12      JZ      CLS120     ;YES - CLEAR IT
5226     1221      FE     CE     .      CPI     EOP        ;END OF DISPLAY?
5227     1223      C8      .      .      RZ      ;YES - RETURN
5228     1224      FE     C4     .      CPI     STPFLG     ;NON-DISPLAYING TERMINATOR?
5229     1226      CC     11     1C      CZ     CHRDL2     ;YES - DELETE IT
5230     1229      C3     16     12      JMP     CLS210     ;CONTINUE SEARCH
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 156
5232	122C	.	.	*****	
5233	122C	.	.	; CLRALL - CLEAR ALL TABS *	
5234	122C	.	.	*****	
5235	122C	.	.	;	
5236	122C	.	.	; ENTRY: H = BASEH	
5237	122C	.	.	;	
5238	122C	.	.	CLRALL EQU \$	
5239	122C	2E	78	MVI L,HTBTBL-BASE ;SET ADDRESS AND NUMBER	
5240	122E	1E	0A	MVI E,HTBLEN ;OF BYTES TO BE CLEARED	
5241	1230	.	.	*****	
5242	1230	.	.	; CLRAL1 - SET A REGION OF RAM TO ZERO *	
5243	1230	.	.	*****	
5244	1230	.	.	;	
5245	1230	.	.	; ENTRY: E = NUMBER OF BYTES IN REGION	
5246	1230	.	.	; H,L = LOW ADDRESS OF REGION	
5247	1230	.	.	;	
5248	1230	.	.	; EXIT : A,E = 0	
5249	1230	.	.	; H,L = H,L(ENTRY) + E	
5250	1230	.	.	;	
5251	1230	.	.	CLRAL1 EQU \$	
5252	1230	AF	.	XRA A ;SET A TO ZERO	
5253	1231	.	.	CLA010 EQU \$	
5254	1231	77	.	MOV M,A ;SET BYTE TO ZERO	
5255	1232	23	.	INX H ;ADVANCE TO NEXT BYTE	
5256	1233	1D	.	DCR E ;ALL BYTES DONE?	
5257	1234	C2	31 12	JNZ CLA010 ;NO - DO NEXT BYTE	
5258	1237	C9	.	RET ;YES - RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 157
5260	1238	.	.	*****	
5261	1238	.	.	; CURPHD - HOME DOWN TO FIRST COLUMN OF *	
5262	1238	.	.	; FIRST LINE BEYOND END OF MEMORY *	
5263	1238	.	.	*****	
5264	1238	.	.	CURPHD EQU \$	
5265	1238	CD	E5 1A	CALL CHKSFK ;DEFINE SOFT KEY MODE?	
5266	1238	C0	.	RNZ ;YES - IGNORE HOME DOWN	
5267	123C	CD	7C 23	CALL CURPRT ;NO - RETURN TO LEFT MARGIN	
5268	123F	.	.	*****	
5269	123F	.	.	; MOVE CURSOR TO NEXT ROW *	
5270	123F	.	.	*****	
5271	123F	.	.	HDC100 EQU \$	
5272	123F	3A	C7 FF	LDA LSTROW	
5273	1242	FE	17 .	CPI MAXROW ;IS LAST ROW DONE AT BOTTOM?	
5274	1244	CC	30 0D	CZ ROLLUP ;YES - ROLL UP THE DISPLAY	
5275	1247	2A	C9 FF	LHLD LSTLIN ;GET CURRENT ROW ADDRESS	
5276	124A	7E	.	MOV A,M ;GET LSB OF NEXT LINE POINTE	
5277	124B	B7	.	ORA A ;IS THERE A NEXT ROW?	
5278	124C	CA	5B 12	JZ HDC200 ;NO - TERMINATE HOME DOWN	
5279	124F	5F	.	MOV E,A ;YES - SET E TO NEXT LINE	
5280	1250	1C	.	INR E ;POINTER OF NEXT LINE	
5281	1251	CD	5D 0D	CALL ROLUP2 ;SET "LSTLIN" AND "CURADR"	
5282	1254	21	C7 FF	LXI H,LSTROW ;TO NEXT LINE	
5283	1257	34	.	INR M ;INCREMENT LAST ROW DONE	
5284	1258	C3	3F 12	JMP HDC100	
5285	125B	.	.	*****	
5286	125B	.	.	; LAST LINE FOUND *	
5287	125B	.	.	; SET ROW	
5288	125B	.	.	*****	
5289	125B	.	.	HDC200 EQU \$	
5290	125B	CD	8F 0C	CALL NXTCHO ;GET 1ST CHAR OF LAST ROW	
5291	125E	FE	CC .	CPI EOL ;LAST ROW EMPTY?	
5292	1260	3A	C7 FF	LDA LSTROW ;(GET LAST ROW POSITION)	
5293	1263	CA	67 12	JZ HDC210 ;YES - SET CURRENT ROW = LAS	
5294	1266	3C	.	INR A ;NO - SET TU NEXT ROW	
5295	1267	.	.	HDC210 EQU \$	
5296	1267	32	C0 FF	STA CURROW ;SET CURRENT ROW NUMBER	
5297	126A	C9	.	RET ;RETURN	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 158
=====
5299     126B      . . .      ;*****
5300     126B      . . .      ; CURPOS - CURSOR POSITIONING *
5301     126B      . . .      ;          INITIAL ENTRY POINT *
5302     126B      . . .      ;*****
5303     126B      . . .      CURPOS EQU $
5304     126B      3A C1 FF    LDA  CURCOL      ;SET NEW COLUMN DEFAULT TO
5305     126E      32 DB FF    STA  NEWCOL      ;CURRENT COLUMN POSITION
5306     1271      3E 7F .     MVI  A,377Q-NRWST
5307     1273      CD 66 05    CALL CLRMF2      ;CLEAR NEW ROW SET FLAG
5308     1276      2E D9 .     MVI  L,SCRNRW-BASE ;PRESET RELATIVE ROW
5309     1278      36 FF .     MVI  M,-1        ;PARAMETER TO -1
5310     127A      21 17 7F    LXI  H,CRPTAB    ;SET RANGE TABLE FOR CURSOR
5311     127D      C3 32 05    JMP  ESCAPA      ;POSITIONING
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 159
5313	1280	.	.	.	;*****	
5314	1280	.	.	.	; NEW COLUMN POSITION IS DEFINED *	
5315	1280	.	.	.	;*****	
5316	1280	.	.	.	CURP01 EQU \$	
5317	1280	0E	4F	.	MVI C,MAXCOL ;SET MAXIMUM VALUE AND	
5318	1282	11	DB	FF	LXI D,NEWCOL ;PARAMETER TO BE SET	
5319	1285	2E	C1	.	MVI L,CURCOL-BASE ;"CHKLIM"	
5320	1287	46	.	.	MOV B,M	
5321	1288	C3	AE	12	JMP CRP025 ;EVALUATE THE PARAMETER	
5322	1288	.	.	.	;*****	
5323	1288	.	.	.	; SCREEN ROW SPECIFIED *	
5324	1288	.	.	.	;*****	
5325	1288	.	.	.	CURP02 EQU \$	
5326	1288	0E	17	.	MVI C,MAXROW ;SET MAXIMUM VALUE AND	
5327	1280	11	D9	FF	LXI D,SCRNRW ;PARAMETER TO BE SET	
5328	1290	2E	C0	.	MVI L,CURROW-BASE ;"CHKLIM"	
5329	1292	46	.	.	MOV B,M	
5330	1293	C3	AE	12	JMP CRP025 ;EVALUATE THE PARAMETER	
5331	1296	.	.	.	;*****	
5332	1296	.	.	.	; NEW ROW POSITION IS DEFINED *	
5333	1296	.	.	.	;*****	
5334	1296	.	.	.	CURP03 EQU \$	
5335	1296	3A	6B	FF	LDA MLKROW ;GET MEMORY LOCK ROW	
5336	1299	87	.	.	ORA A ;MEMORY LOCK ENABLED?	
5337	129A	C2	B1	12	JNZ CRP050 ;YES - IGNORE PARAMETER	
5338	129D	3E	80	.	MVI A,NWRWST ;NO - SET NEW ROW SET	
5339	129F	CD	94	18	CALL SETMF2 ;FLAG	
5340	12A2	0E	FF	.	MVI C,255 ;SET MAXIMUM VALUE AND	
5341	12A4	11	DA	FF	LXI D,NEWROW ;PARAMETER TO BE SET	
5342	12A7	3A	C0	FF	LDA CURROW ;COMPUTE CURRENT ABSOLUTE	
5343	12AA	2E	A3	.	MVI L,TLINO ;ROW ADDRESS	
5344	12AC	86	.	.	ADD M	
5345	12AD	47	.	.	MOV B,A ;PUT IT INTO B-REGISTER	
5346	12AE	.	.	.	CRP025 EQU \$	
5347	12AE	CD	3C	11	CALL CHKLI0 ;EVALUATE INPUT PARAMETER	
5348	12B1	.	.	.	CRP050 EQU \$	
5349	12B1	3A	88	FF	LDA CHAR ;RECALL THE INPUT CHARACTER	
5350	12B4	E6	20	.	ANI 40Q ;IS IT AN UPPER CASE CHAR?	
5351	12B6	C2	3A	05	JNZ ESCAPB ;NO - CONTINUE ESC SEQUENCE	
5352	12B9	.	.	.	; YES - POSITION CURSOR	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 160
=====
5354     12B9     . . .      ;*****
5355     12B9     . . .      ; EXECUTE COMPLETED SEQUENCE *
5356     12B9     . . .      ;*****
5357     12B9     3A D9 FF      LDA  SCNRW  ;GET SCREEN ROW PARAMETER
5358     12BC     87 . .      ORA  A      ;WAS SCREEN ROW ADDRESS SET?
5359     12BD     FA C3 12      JM  CRP200  ;NO - SET ABSOLUTE ROW ADDR
5360     12C0     32 C0 FF      STA  CURROW ;YES - SET NEW SCREEN ROW
5361     12C3     . . .      ;*****
5362     12C3     . . .      ; SET ABSOLUTE ROW ADDRESS *
5363     12C3     . . .      ;*****
5364     12C3     . . .      CRP200 EQU $
5365     12C3     FC D7 12      CM   CRP500 ;FIND LOCATION OF NEW ROW
5366     12C6     . . .      ;
5367     12C6     . . .      ; SET COLUMN ADDRESS
5368     12C6     . . .      ;
5369     12C6     3A DB FF      LDA  NEWCOL ;GET NEW COLUMN ADDRESS
5370     12C9     . . .      ;*****
5371     12C9     . . .      ; LOCATE ADDRESS OF CHARACTER *
5372     12C9     . . .      ;*****
5373     12C9     . . .      CURP04 EQU $
5374     12C9     32 C1 FF      STA  CURCOL ;STORE NEW COLUMN ADDRESS
5375     12CC     CD 0D 08      CALL RCADDR ;FIND CHARACTER
5376     12CF     C8 . .      RZ      ;CHARACTER FOUND - RETURN
5377     12D0     . . .      ;*****
5378     12D0     . . .      ; CHARACTER NOT CURRENTLY STORED *
5379     12D0     . . .      ; BUILD LINE OVER TO NEW POSITION *
5380     12D0     . . .      ;*****
5381     12D0     2E 89 .      MVI  L,DCHAR ;SET UP BLANK FOR NEW POS.
5382     12D2     36 20 .      MVI  M,ABLNK
5383     12D4     C3 0E 25      JMP  DISPLO  ;BUILD BLOCKS
=====

```

13255
2648A MICROCODE LISTING 'PT91'

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
5385	12D7	.	.	*****
5386	12D7	.	.	; LOCATE NEW ABSOLUTE ROW LOCATION *
5387	12D7	.	.	*****
5388	12D7	.	.	CRP500 EQU \$
5389	12D7	3A	6F FF	LDA MFLGS2 ;GET TERMINAL MODE FLAGS
5390	12DA	E6	80 .	ANI NWRWST ;NEW ABSOLUTE ROW SET?
5391	12DC	C8	. .	RZ ;NO - RETURN
5392	12DD	3A	DA FF	LDA NEWROW ;GET NEW ROW VALUE
5393	12E0	2E	A3 .	MVI L,TLINO ;SUBTRACT ROW CORRESP.
5394	12E2	96	. .	SUB M ;TO TOP OF PAGE
5395	12E3	2E	C0 .	MVI L,CURROW
5396	12E5	DA	BD 0C	JC PRVPG1 ;LOCATE PREVIOUS ROW PAGE
5397	12E8	FE	18 .	CPI MAXROW+1
5398	12EA	77	. .	MOV M,A ;SET NEW ROW
5399	12EB	D8	. .	RC ;RETURN IF SAME PAGE
5400	12EC	.	.	*****
5401	12EC	.	.	; ROW IS AFTER BOTTOM OF PAGE *
5402	12EC	.	.	; ROLL DISPLAY UP *
5403	12EC	.	.	*****
5404	12EC	36	17 .	MVI M,MAXROW ;SET ROW
5405	12EE	D6	17 .	SUI MAXROW ;SET ROLL COUNT
5406	12F0	.	.	STR010 EQU \$
5407	12F0	CD	4E 0C	CALL NXTPG1 ;ROLL DISPLAY UP
5408	12F3	7E	. .	MOV A,M ;GET NUMBER OF ROWS TO ROLL
5409	12F4	91	. .	SUB C ;SUBTRACT ROWS ROLLED
5410	12F5	C8	. .	RZ ;RETURN IF ROLL COMPLETE
5411	12F6	77	. .	MOV M,A ;SAVE NUMBER OF ROW TO ROLL
5412	12F7	AF	. .	XRA A ;(SET TO FIND COLUMN 0)
5413	12F8	CD	10 08	CALL RCADRO ;BUILD NEW ROWS
5414	12F8	C0	. .	RNZ ;RETURN IF OUT OF MEMORY
5415	12FC	3A	83 FF	LDA NMROLL ;GET # OF ROWS TO ROLL
5416	12FF	C3	F0 12	JMP STR010 ;ROLL AGAIN
5417	1302	.	.	*****
5418	1302	.	.	; CURSEN - CURSOR POSITION SENSE *
5419	1302	.	.	*****
5420	1302	.	.	;
5421	1302	.	.	; RLCRSN - SCREEN RELATIVE CURSOR SENSE
5422	1302	.	.	;
5423	1302	.	.	RLCRSN EQU \$
5424	1302	3E	04 .	MVI A,RELSNS ;SET RELATIVE SENSE FLAG
5425	1304	CD	94 18	CALL SETMF2
5426	1307	C3	0F 13	JMP CUR100 ;GO SET CURSOR SENSE FLAG
5427	130A	.	.	;
5428	130A	.	.	; CURSEN - ABSOLUTE CURSOR SENSE
5429	130A	.	.	;
5430	130A	.	.	CURSEN EQU \$
5431	130A	3E	FB .	MVI A,3770-RELSNS
5432	130C	CD	66 05	CALL CLRMF2 ;CLEAR RELATIVE SENSE FLAG
5433	130F	.	.	CUR100 EQU \$
5434	130F	01	00 10	LXI B,SCRSEN ;SET UP BLOCK TRANSFER

13255
2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 162
=====
5435     1312     C3  25  18          JMP  SBLXF0          ;FOR CURSOR SENSE PENDING
=====
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 163
5437	1315	.	.	CRSNGO EQU \$	
5438	1315	01	FF	EF LXI B,-1-SCRSEN ;CLEAR CURSOR SENSE	
5439	1318	CD	98	11 CALL CLBLXF ;PENDING FLAG	
5440	1318	06	26	. MVI B,AMPSND ;SEND <ESC>--<&>	
5441	131D	CD	1C	19 CALL ESCOUT	
5442	1320	3E	61	. MVI A,SMALLA ;TRANSMIT LOWER CASE A	
5443	1322	CD	22	19 CALL XPUTDC	
5444	1325	.	.	. ;*****	
5445	1325	.	.	. ; OUTPUT CURSOR COLUMN *	
5446	1325	.	.	. ;*****	
5447	1325	21	22	19 LXI H,XPUTDC ;SEND NUMBER TO DATA COMM	
5448	1328	3A	C1	FF LDA CURCOL ;GET CURRENT CURSOR COLUMN	
5449	1328	CD	23	09 CALL BN2DE1 ;CONVERT AND TRANSMIT VALUE	
5450	132E	3E	63	. MVI A,ALCC ;TRANSMIT LOWER CASE C	
5451	1330	CD	22	19 CALL XPUTDC	
5452	1333	.	.	. ;*****	
5453	1333	.	.	. ; OUTPUT CURSOR ROW *	
5454	1333	.	.	. ;*****	
5455	1333	3A	6F	FF LDA MFLGS2 ;GET TERMINAL MODE FLAGS	
5456	1336	E6	04	. ANI RELSNS ;SCREEN RELATIVE SENSING?	
5457	1338	3A	C0	FF LDA CURROW ;(GET CURSOR ROW NUMBER)	
5458	1338	06	59	. MVI B,Y ;(SET DEFAULT PARAMETER)	
5459	133D	C2	46	13 JNZ CRS100 ;YES - OUTPUT SCREEN ADDRESS	
5460	1340	21	A3	FF LXI H,TLIND ;NO - COMPUTE ABSOLUTE	
5461	1343	86	.	. ADD M ;ROW NUMBER	
5462	1344	06	52	. MVI B,R ;SET ABSOLUTE PARAMETER CHAR	
5463	1346	.	.	. ;*****	
5464	1346	.	.	. ; TRANSMIT ROW PARAMETER *	
5465	1346	.	.	. ;*****	
5466	1346	.	.	. ;	
5467	1346	.	.	. ; A = ROW VALUE	
5468	1346	.	.	. ; B = ROW PARAMETER LETTER	
5469	1346	.	.	. ;	
5470	1346	.	.	. CRS100 EQU \$	
5471	1346	C5	.	. PUSH B ;SAVE ROW PARAMETER LETTER	
5472	1347	CD	26	09 CALL BN2DE2 ;CONVERT AND TRANSMIT VALUE	
5473	134A	F1	.	. POP PSW ;RECALL ROW PARAMETER LETTER	
5474	1348	CD	22	19 CALL XPUTDC ;TRANSMIT ROW PARAMETER CHAR	
5475	134E	.	.	. ; FALL INTO "SDTERM"	
5476	134E	.	.	. ;*****	
5477	134E	.	.	. ; SDTERM - SEND BLOCK TERMINATOR *	
5478	134E	.	.	. ; RS IF PAGE MODE, OTHERWISE CR(LF) *	
5479	134E	.	.	. ;*****	
5480	134E	.	.	. SDTERM EQU \$	
5481	134E	CD	51	18 CALL SDTRM1 ;SEND TERMINATOR	
5482	1351	.	.	. SDTER1 EQU \$	
5483	1351	CD	89	11 CALL CLRFXN ;CLEAR BLOCK TERMINATOR	
5484	1354	3E	07	. MVI A,ENDBLK ;TELL DATA COMM THAT LAST	
5485	1356	C3	73	13 JMP DCMCTL ;CHARACTER IN BLOCK IS OUT	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 164
=====
5487     1359      . . .      ;
5488     1359      . . .      ; DC260 - OUTPUT DC2
5489     1359      . . .      ;
5490     1359      . . .      DC260 EQU $
5491     1359     21 70 FF      LXI H,MFLGS
5492     135C     7E . .      MOV A,M ;CLEAR DC2 PENDING FLAG
5493     135D     E6 FE .      ANI -1-SDC2/256
5494     135F     77 . .      MOV M,A
5495     1360     3E 0D .      MVI A,PROMPT ;TELL DATA COMM ROUTINE TO
5496     1362     C3 73 13     JMP DCMCTL ;SEND PROMPT CODE
5497     1365      . . .      ;*****
5498     1365      . . .      ; DCMINT - DATA COMM INTERRUPT ROUTINE *
5499     1365      . . .      ;*****
5500     1365      . . .      ;
5501     1365      . . .      ; ENTRY: PSW "PUSHED"
5502     1365      . . .      ; A = INTERRUPT CODE
5503     1365      . . .      ;
5504     1365      . . .      DCMINT EQU $
5505     1365     CD 65 91     CALL INTVEC ;CHECK ALTERNATE INTERRUPT
5506     1368     F1 . .      POP PSW ;RESTORE PSW AND A-REGISTER
5507     1369     C3 26 50     JMP ZDCINT ;EXECUTE NORMAL DATA COMM
5508     136C      . . .      ; INTERRUPT ROUTINE
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 165
5510	136C	.	.	*****	
5511	136C	.	.	; BRKDC - EXECUTE DATA COMM BREAK *	
5512	136C	.	.	*****	
5513	136C	.	.	BRKDC EQU \$	
5514	136C	3E	05	MVI A,PUTBRK ;EXECUTE DATACOM BREAK	
5515	136E	C3	73 13	JMP DCMCTL ;CONTROL	
5516	1371	.	.	;	
5517	1371	.	.	; DISMDM - DISCONNECT MODEM	
5518	1371	.	.	;	
5519	1371	.	.	DISMDM EQU \$	
5520	1371	3E	06	MVI A,DISCNT ;EXECUTE MODEM DISCONNECT	
5521	1373	.	.	*****	
5522	1373	.	.	; DCMCTL - PERFORM DATA COMM CONTROL FUNCTION *	
5523	1373	.	.	*****	
5524	1373	.	.	;	
5525	1373	.	.	; ENTRY: A = FUNCTION TYPE NUMBER	
5526	1373	.	.	;	
5527	1373	.	.	; EXIT: Z - FUNCTION PERFORMED	
5528	1373	.	.	; NZ - FUNCTION NOT PERFORMED	
5529	1373	.	.	;	
5530	1373	.	.	DCMCTL EQU \$	
5531	1373	F5	.	PUSH PSW ;SAVE A-REGISTER	
5532	1374	CD	95 11	CALL CKRMTE ;REMOTE MODE ENABLED?	
5533	1377	C2	7D 13	JNZ DCC010 ;YES - PERFORM FUNCTION	
5534	137A	F1	.	POP PSW ;NO - RESTORE A-REGISTER	
5535	137B	3C	.	INR A ;FORCE NZ	
5536	137C	C9	.	RET ;RETURN	
5537	137D	.	.	;	
5538	137D	.	.	DCC010 EQU \$	
5539	137D	F1	.	POP PSW ;RESTORE A-REGISTER	
5540	137E	.	.	DCMCT1 EQU \$;ENTRY TO FORCE DATA COMM CT	
5541	137E	CD	11 50	CALL ZDCCTL ;EXECUTE FUNCTION	
5542	1381	D0	.	RNC ;SUCCESSFUL - RETURN	
5543	1382	.	.	DCERR EQU \$;PROCESS DATA COMM ERROR	
5544	1382	CA	14 48	JZ ZBELL ;NOT FATAL - SOUND BELL	
5545	1385	.	.	*****	
5546	1385	.	.	; DISPLAY TEST FAIL MESSAGES *	
5547	1385	.	.	*****	
5548	1385	.	.	HANGU0 EQU \$	
5549	1385	CD	2F 1E	CALL DSPMS0 ;DISPLAY THE ERROR MESSAGE	
5550	1388	.	.	HANGUP EQU \$	
5551	1388	3E	04	MVI A,FRCRST ;SET TO FORCE FULL RESET	
5552	138A	CD	44 15	CALL STCMFL ;IF RESET KEY HIT	
5553	138D	.	.	;	
5554	138D	.	.	HNG010 EQU \$	
5555	138D	CD	DA 10	CALL DISLN4 ;RE-ENABLE RESET ONLY	
5556	1390	C3	8D 13	JMP HNG010 ;HANG TERMINAL	
5557	1393	.	.	;	*****
5558	1393	.	.	;	* RESET KEY MUST BE HIT *
5559	1393	.	.	;	* TO RESTORE TERMINAL *

13255
2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 166
=====
5560     1393      . . . ;                                     * OPERATION *
5561     1393      . . . ;                                     *****
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 167
5563	1393	.	.	. ;*****	
5564	1393	.	.	. ; DCNUM - ACCUMULATE PARAMETER FOR ESC SEQ *	
5565	1393	.	.	. ;*****	
5566	1393	.	.	. ;	
5567	1393	.	.	. ; EXIT : Z TRUE	
5568	1393	.	.	. ;	
5569	1393	.	.	. DCNUM EQU \$	
5570	1393	21	DD	FF LXI H,IOCSGN ;GET THE CURRENT SIGN	
5571	1396	7E	.	. MOV A,M ;VALUE	
5572	1397	87	.	. ORA A ;HAS ANY SIGN BEEN SET?	
5573	1398	C2	9D	13 JNZ DCN005 ;YES - DON'T CHANGE IT	
5574	1398	36	80	. MVI M,NUSIGN ;NO - SET NO SIGN FLAG	
5575	139D	.	.	. DCN005 EQU \$	
5576	139D	3A	88	FF LDA CHAR ;GET INPUT CHARACTER	
5577	13A0	D6	30	. SUI ZERO ;EXTRACT BINARY VALUE	
5578	13A2	5F	.	. MOV E,A ;PUT VALUE IN E-REGISTER	
5579	13A3	16	00	. MVI D,0 ;SET MSB TO ZERO	
5580	13A5	3A	D4	FF LDA RADIX ;GET RADIX OF NUMBER	
5581	13A8	2A	DE	FF LHLD IODATA ;GET ACCUMULATOR	
5582	13AB	EB	.	. XCHG ;PUT ACCUMULATOR IN D,E	
5583	13AC	.	.	. ;	
5584	13AC	.	.	. DCN010 EQU \$	
5585	13AC	19	.	. DAD D ;ACCUMULATE NEW VALUE	
5586	13AD	3D	.	. DCR A ;RADIX ADJUSTMENT COMPLETED?	
5587	13AE	C2	AC	13 JNZ DCN010 ;NO - CONTINUE ADDING	
5588	13B1	22	DE	FF SHLD IODATA ;YES - STORE NEW VALUE	
5589	13B4	C3	42	05 JMP ESCAP1 ;CONTINUE ESCAPE SEQUENCE	
5590	13B7	.	.	. ;*****	
5591	13B7	.	.	. ; DCPLUS - PLUS SIGN RECEIVED FOR PARAMETER *	
5592	13B7	.	.	. ;*****	
5593	13B7	.	.	. DCPLUS EQU \$	
5594	13B7	06	01	. MVI B,1 ;SET B-REG TO SIGN VALUE	
5595	13B9	C3	BE	13 JMP DCM010 ;SET SIGN FLAG	
5596	13BC	.	.	. ;*****	
5597	13BC	.	.	. ; DCMNUS - MINUS SIGN RECEIVED FOR PARAMETER *	
5598	13BC	.	.	. ;*****	
5599	13BC	.	.	. DCMNUS EQU \$	
5600	13BC	06	FF	. MVI B,-1	
5601	13BE	.	.	. DCM010 EQU \$	
5602	13BE	21	DD	FF LXI H,IOCSGN ;GET CURRENT SIGN VALUE	
5603	13C1	7E	.	. MOV A,M	
5604	13C2	87	.	. ORA A ;SIGN SET ALREADY?	
5605	13C3	C2	48	05 JNZ ESCEND ;YES - ABORT ESCAPE SEQUENCE	
5606	13C6	70	.	. MOV M,B ;NO - SET SIGN VALUE	
5607	13C7	C3	42	05 JMP ESCAP1 ;CONTINUE ESCAPE SEQUENCE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 168
=====
5609     13CA      . . .      ;*****
5610     13CA      . . .      ; DCXTEST - EXECUTE DATA COMM SELF-TEST *
5611     13CA      . . .      ;*****
5612     13CA      . . .      DCXTEST EQU $
5613     13CA      CD 95 11    CALL CKRMTE      ;REMOTE MODE ENABLED?
5614     13CD      C8 . .     RZ              ;NO - DON'T DO SELF-TEST
5615     13CE      CD 14 50    CALL ZDCTST     ;CALL DATA COMM SELF-TEST
5616     13D1      DA 85 13    JC HANGUO      ;HANG TERMINAL IF FATAL ERRO
5617     13D4      C3 30 1E    JMP DSPMS1     ;DISPLAY MESSAGE AND EXIT
5618     13D7      . . .      ;              IF SELF-TEST SUCCESSFUL
5619     13D7      . . .      ;*****
5620     13D7      . . .      ; DCXB2D - SEE IF SOURCE OF CHARACTER IS *
5621     13D7      . . .      ;   DATA COMM OR I/O BUFFER          *
5622     13D7      . . .      ;*****
5623     13D7      . . .      ;
5624     13D7      . . .      ; EXIT :  Z - INPUT IS NOT FROM DATA COMM OR I/O
5625     13D7      . . .      ;           NZ - INPUT IS FROM DATA COMM OR I/O
5626     13D7      . . .      ;           A DESTROYED
5627     13D7      . . .      ;
5628     13D7      . . .      DCXB2D EQU $
5629     13D7      3A 6E FF    LDA DFLGS      ;GET DATA TRANSFER FLAGS
5630     13DA      E6 81 .     ANI SDACOM+XBF2DS ;SET Z-FLAG
5631     13DC      C9 . .     RET           ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 169
5633	13DD	.	.	. ;*****	
5634	13DD	.	.	. ; DELAY0 - PAUSE FOR 1 SECOND *	
5635	13DD	.	.	. ;*****	
5636	13DD	.	.	. DELAY0 EQU \$	
5637	13DD	3E	18	. MVI A,MAXROW+1 ;REMOVE CURSOR AND	
5638	13DF	CD	D6	10 CALL DISLN2 ;RE-ENABLE RESET KEY	
5639	13E2	2E	64	. MVI L,100 ;DELAY FOR 1 SECOND	
5640	13E4	.	.	. ;*****	
5641	13E4	.	.	. ; DELAY - DELAY 10 MILLISECONDS *	
5642	13E4	.	.	. ; TIMES COUNT IN L *	
5643	13E4	.	.	. ;*****	
5644	13E4	.	.	. DELAY EQU \$	
5645	13E4	3E	00	. MVI A,SETROM	
5646	13E6	D3	70	. OUT PROCSR ;RESET THE TIMER	
5647	13E8	3A	F5	FF LDA PRCCTL ;RESTORE PROCESSOR STATE	
5648	13EB	D3	70	. OUT PROCSR	
5649	13ED	.	.	. ;	
5650	13ED	.	.	. DLY010 EQU \$	
5651	13ED	AF	.	. XRA A ;CLEAR THE INTERRUPT FLAG	
5652	13EE	32	F6	FF STA INTFLG	
5653	13F1	.	.	. DLY020 EQU \$	
5654	13F1	76	.	. HLT ;SLEEP UNTIL INTERRUPTED	
5655	13F2	3A	F6	FF LDA INTFLG ;GET INTERRUPT FLAG	
5656	13F5	FE	03	. CPI TMRINT ;TIMER INTERRUPT?	
5657	13F7	C2	F1	13 JNZ DLY020 ;NO - CONTINUE WAITING	
5658	13FA	2D	.	. DCR L ;ENOUGH TIMER INTERRUPTS?	
5659	13FB	C2	ED	13 JNZ DLY010 ;NO - CONTINUE TIMING	
5660	13FE	C9	.	. RET ;YES - RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 170
=====
5662     13FF      .      .      .      ;*****
5663     13FF      .      .      .      ; <F> - SEND FUNCTION DATA *
5664     13FF      .      .      .      ;*****
5665     13FF      .      .      .      SNDCD2 EQU $
5666     13FF      3A     DE     FF      LDA  IODATA      ;GET ACCUMULATED VALUE
5667     1402      47     .      .      MOV  B,A         ;PUT CODE INTO B-REGISTER
5668     1403      3E     0C     .      MVI  A,SNDFCT   ;SET DATA COMM CONTROL CODE
5669     1405      CD     73    13     CALL DCMCTL     ;PERFORM FUNCTION
5670     1408      C0     .      .      RNZ              ;EXIT IF FUNCTION NOT DONE
5671     1409      .      .      .      ;              OTHERWISE, SEND SCREEN DATA
5672     1409      .      .      .      ;*****
5673     1409      .      .      .      ; DISPLAY SEND *
5674     1409      .      .      .      ;*****
5675     1409      .      .      .      DPSSEND EQU $
5676     1409      3E     08     .      MVI  A,CKIOKY
5677     140B      CD     08    48     CALL ZKBCTL     ;I/O CONTROL KEY DOWN ALSO?
5678     140E      C2     CA    13     JNZ  DCTEST     ;YES - DO DATA COMM SELF-TEST
5679     1411      3A     F8    FF     LDA  CMFLGS     ;GET COMMON FLAGS
5680     1414      E6     10     .      ANI  REMSET     ;REMOTE ENABLED?
5681     1416      11     0A    00     LXI  D,RCKYCD-ENTRCD*2 ;(SET KEY INDEX)
5682     1419      CA     DE    16     JZ   IOKEYS     ;NO - PERFORM RECORD COMMAND
5683     141C      01     00    40     LXI  B,SENDER   ;YES - SET XFR PENDING FLAG
5684     141F      3A     F3    FF     LDA  MDFLG2     ;NO - GET TERMINAL MODE FLAG
5685     1422      E6     02     .      ANI  BLKMODE    ;BLOCK MODE ENABLED?
5686     1424      CA     3F    14     JZ   DPS200     ;NO - DO CHARACTER MODE STAR
5687     1427      CD     E5    1A     CALL CHKSFK     ;SOFT KEY MODE?
5688     142A      C2     57    14     JNZ  DPS220     ;YES - DON'T SET TERMINATOR
5689     142D      3A     FA    FF     LDA  KBJMP2     ;YES - GET KEYBOARD JUMPERS
5690     1430      E6     01     .      ANI  AUTTRM     ;AUTO TERMINATE ENABLED?
5691     1432      CA     57    14     JZ   DPS220     ;NO - DO DON'T MOVE CURSOR
5692     1435      CD     D2    18     CALL STTERM     ;YES - SET NON-DISPLAYING
5693     1438      .      .      .      ;              TERMINATOR
5694     1438      C8     .      .      RZ              ;EXIT IF NOT SUCCESSFUL
5695     1439      .      .      .      ;*****
5696     1439      .      .      .      ; FIRST TRANSMIT CHARACTER LOCATED - SET *
5697     1439      .      .      .      ;   TRANSFER PENDING FLAG                               *
5698     1439      .      .      .      ;*****
5699     1439      .      .      .      DPS100 EQU $
5700     1439      01     00    40     LXI  B,SENDER   ;SET B,C XFR PENDING FLAG
5701     143C      C3     30    18     JMP  SBLXF1     ;FOR BLOCK MODE TRANSFER
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 171
5703	143F	.	.	. ;*****	
5704	143F	.	.	. ; AUTO TERMINATOR JUMPER NOT REMOVED - DO *	
5705	143F	.	.	. ; NORMAL DATA ENTRY FROM DISPLAY *	
5706	143F	.	.	. ;*****	
5707	143F	.	.	. DPS200 EQU \$	
5708	143F	3A	FB	FF LDA KBJMPR ;GET KEYBOARD JUMPERS 1	
5709	1442	E6	C0	. ANI HNDSHK+DC2SND	
5710	1444	EE	40	. XRI HNDSHK ;HANDSHAKE ENABLED?	
5711	1446	C4	CD	04 CNZ CHKCT1 ;NO - SET BLOCK TRIGGER	
5712	1449	.	.	. DPS210 EQU \$	
5713	1449	CD	28	18 CALL SBLXFA ;SET TRANSFER PENDING FLAG	
5714	144C	.	.	. DPS215 EQU \$	
5715	144C	CD	CF	1A CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE	
5716	144F	C0	.	. RNZ ;YES - DON'T MOVE CURSOR	
5717	1450	.	.	. ;*****	
5718	1450	CD	23	60 CALL ZMUCHK ;AUTO PLOT MENU UP?	
5719	1453	C0	.	. RNZ ;YES, DONT MOVE CURSOR	
5720	1454	.	.	. ;*****	
5721	1454	C3	7F	23 JMP CRRET1 ;NO - PUT CURSOR AT BEGINNIN	
5722	1457	.	.	. ; OF LINE (A = 0)	
5723	1457	.	.	. ;	
5724	1457	.	.	. ; SET KEYBOARD BLOCK TRANSFER	
5725	1457	.	.	. ;	
5726	1457	.	.	. DPS220 EQU \$	
5727	1457	CD	30	18 CALL SBLXF1 ;SET BLOCK MODE XFR PENDING	
5728	145A	3A	70	FF LDA MFLGS ;GET TRANSFER PENDING FLAGS	
5729	145D	E6	01	. ANI SDC2/256 ;DC2 TO BE SENT?	
5730	145F	C0	.	. RNZ ;YES - DON'T MOVE CURSOR	
5731	1460	CD	8A	11 CALL CKLNMD ;LINE MODE?	
5732	1463	CA	4C	14 JZ DPS215 ;YES - SET CURSOR IN LINE	
5733	1466	.	.	. ;*****	
5734	1466	.	.	. ; DPSEN1 - HOME CURSOR FOR TRANSMISSION *	
5735	1466	.	.	. ;*****	
5736	1466	.	.	. DPSEN1 EQU \$	
5737	1466	.	.	. ;*****	
5738	1466	CD	23	60 CALL ZMUCHK ;AUTO PLOT MENU ON?	
5739	1469	C2	6A	60 JNZ ZAPHME ;YES, HOME MENU CURSOR	
5740	146C	.	.	. ;*****	
5741	146C	CD	55	19 CALL XMOHME ;HOME CURSOR	
5742	146F	CD	CF	1A CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE	
5743	1472	C0	.	. RNZ ;YES - RETURN	
5744	1473	C3	7F	23 JMP CRRET1 ;NO - SET CURSOR TO COLUMN 0	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 172
5746	1476	.	.	.	;
5747	1476	.	.	.	; * * * * *
5748	1476	.	.	.	;
5749	1476	.	.	.	; DPSGO - SEND DISPLAY TO DATACOM
5750	1476	.	.	.	;
5751	1476	.	.	.	; ENTRY: CURCOL,CURROW SET TO STARTING
5752	1476	.	.	.	LOCATION
5753	1476	.	.	.	;
5754	1476	.	.	.	; EXIT : ALL REGISTERS DESTROYED
5755	1476	.	.	.	;
5756	1476	.	.	.	DPSGO EQU \$
5757	1476	CD	2E	7D	CALL INITDG ;INIT DISPLAY GET ROUTINE
5758	1479	C2	C8	14	JNZ DSG200 ;TERMINATE IF NO CHARACTERS
5759	147C	3E	FF	.	MVI A,STPXR ;SET TERMINATOR FUNCTION TO
5760	147E	32	6D	FF	STA TRMFC ;TERMINATE TRANSFER
5761	1481	.	.	.	DSG010 EQU \$
5762	1481	3E	0D	.	MVI A,STCHST ;SET CHARACTER SET FOR
5763	1483	CD	08	48	CALL ZKBCTL ;FOREIGN TERMINALS?
5764	1486	DC	22	19	CC XPUTDC ;YES - OUTPUT SI/SO
5765	1489	.	.	.	;
5766	1489	.	.	.	; OUTPUT CHARACTERS FROM DISPLAY
5767	1489	.	.	.	;
5768	1489	.	.	.	DSG020 EQU \$
5769	1489	CD	A3	26	CALL GETDSP ;ANY CHARACTER?
5770	148C	DA	98	14	JC DSG100 ;NO - CHECK TERMINATION
5771	148F	CD	22	19	CALL XPUTDC ;YES - TRANSMIT THE CHARACTE
5772	1492	D2	89	14	JNC DSG020 ;CONTINUE IF NO DATA COMM ER
5773	1495	C3	F5	14	JMP DSG230 ;ELSE, TERMINATE OUTPUT
5774	1498	.	.	.	;
5775	1498	.	.	.	; NON-CHARACTER FOUND - CHECK TERMINATION
5776	1498	.	.	.	;
5777	1498	.	.	.	DSG100 EQU \$
5778	1498	FA	C8	14	JM DSG200 ;END OF DISPLAY - TERMINATE
5779	149B	47	.	.	MOV B,A ;SAVE EXIT STATUS
5780	149C	CD	84	11	CALL GTMODE ;PAGE MODE ENABLED?
5781	149F	CA	D7	14	JZ DSG210 ;NO - END WITH CR(LF)
5782	14A2	CD	D4	1A	CALL CHKFMT ;FORMAT MODE?
5783	14A5	CA	B1	14	JZ DSG110 ;NO - SEND CR AND LF
5784	14A8	3A	03	50	LDA RECSEP ;YES - END WITH RECORD
5785	14AB	CD	22	19	CALL XPUTDC ;SEPARATOR
5786	14AE	C3	81	14	JMP DSG010 ;CONTINUE THRU DISPLAY

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 173
=====
5788     14B1      . . .      ;
5789     14B1      . . .      ; EOL FOR NON-FORMAT PAGE MODE - SEND CR AND LF
5790     14B1      . . .      ;
5791     14B1      . . .      DSG110 EQU $
5792     14B1      3E 0D      MVI A,CR      ;SEND RETURN
5793     14B3      CD 22 19    CALL XPUTDC    ;AND
5794     14B6      CD 65 18    CALL SDTRM3    ;LINE FEED
5795     14B9      CD E5 1A    CALL CHKSFK    ;SOFT KEY DEFINE MODE?
5796     14BC      . . .      ;*****
5797     14BC      CC 23 60    CZ ZMUCHK     ;NO--IS AUTOPLLOT MENU UP?
5798     14BF      CC 6F 0B    CZ LNFEED     ;NEITHER IS ON, DO LINE FEED
5799     14C2      . . .      ;*****
5800     14C2      CD D3 10    CALL DISLN1    ;SET DISPLAY CURSOR ROW
5801     14C5      C3 81 14    JMP DSG010     ;CONTINUE THRU DISPLAY
5802     14C8      . . .      ;
5803     14C8      . . .      ; END OF DISPLAY - SEND TERMINATOR
5804     14C8      . . .      ;
5805     14C8      . . .      DSG200 EQU $
5806     14C8      3A 04 50    LDA BLKTRM    ;SEND BLOCK TERMINATOR
5807     14CB      CD 22 19    CALL XPUTDC    ;CHARACTER
5808     14CE      CD 84 11    CALL GTMODE    ;PAGE MODE?
5809     14D1      CA EB 14    JZ DSG220     ;NO - SEND CR(LF)
5810     14D4      C3 EE 14    JMP DSG225     ;YES - CLEAR XFR PENDING FLA
5811     14D7      . . .      ;*****
5812     14D7      . . .      ; NON-PAGE MODE TERMINATION - SEND CR(LF) *
5813     14D7      . . .      ;*****
5814     14D7      . . .      DSG210 EQU $
5815     14D7      CD CF 1A    CALL CHKFMS    ;FORMAT/SOFT KEY MODE?
5816     14DA      C2 EB 14    JNZ DSG220     ;YES - DON'T DO LINE FEED
5817     14DD      . . .      ;*****
5818     14DD      CD 23 60    CALL ZMUCHK    ;AUTOPLLOT MENU BEING SENT?
5819     14E0      C2 EB 14    JNZ DSG220     ;YES, DONT DO LF
5820     14E3      . . .      ;*****
5821     14E3      3A F3 FF    LDA MDFLG2    ;NO - GET SOFT MODE FLAGS
5822     14E6      E6 04      ANI AUTOLF    ;AUTO LINE FEED ENABLED?
5823     14E8      C4 6F 0B    CNZ LNFEED    ;YES - DO LINE FEED
5824     14EB      . . .      ;*****
5825     14EB      . . .      ; SEND CR(LF) TERMINATOR *
5826     14EB      . . .      ;*****
5827     14EB      . . .      DSG220 EQU $
5828     14EB      CD 51 18    CALL SDTRM1    ;SEND CR(LF)
5829     14EE      . . .      DSG225 EQU $
5830     14EE      CD 51 13    CALL SDTER1    ;MARK END OF OUTPUT BLOCK
5831     14F1      AF . .      XRA A          ;RESET TERMINATOR FUNCTION
5832     14F2      32 6D FF    STA TRMFACT    ;TO DELETE TERMINATOR
5833     14F5      . . .      DSG230 EQU $
5834     14F5      01 FF BF    LXI B,-1-SENER
5835     14F8      CD 9B 11    CALL CLBLXF    ;CLEAR ENTER PENDING FLAG
5836     14FB      C3 AF 21    JMP CRADV1     ;CLEAR CURSOR ADVANCE FLAG
5837     14FE      . . .      ; AND EXIT
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 174
=====
5839     14FE      . . .      ;*****
5840     14FE      . . .      ; A2OUTB - PUT BYTE INTO OUTPUT BUFFER *
5841     14FE      . . .      ;*****
5842     14FE      . . .      ; ENTRY:  A = BYTE TO BE OUTPUT
5843     14FE      . . .      ;
5844     14FE      . . .      ; EXIT :  H = BASEH
5845     14FE      . . .      ;          B2DEND = B2DEND + 1
5846     14FE      . . .      ;          D,E,L DESTROYED
5847     14FE      . . .      ;
5848     14FE      . . .      ; ECOUTB - OUTPUT <ESC>
5849     14FE      . . .      ;
5850     14FE      . . .      ; ENTRY:  B = SECOND CHARACTER IN ESCAPE SEQ
5851     14FE      . . .      ;
5852     14FE      . . .      ECOUTB EQU $
5853     14FE      3E 18 .      MVI  A,ESC      ;SET A TO ESC
5854     1500      CD 04 15      CALL A2OUTB     ;PUT ESC INTO OUTPUT BUFFER
5855     1503      . . .      B2OUTB EQU $
5856     1503      78 . . .      MOV  A,B        ;PUT SECOND CHAR INTO A-REG
5857     1504      . . .      ; FALL INTO OUTPUT ROUTINE
5858     1504      . . .      ;
5859     1504      . . .      A2OUTB EQU $
5860     1504      21 38 FF      LXI  H,B2DEND
5861     1507      34 . . .      INR  M          ;INCREMENT TO NEXT POSITION
5862     1508      6E . . .      MOV  L,M        ;GET NEW ADDRESS
5863     1509      77 . . .      MOV  M,A        ;STORE THE BYTE
5864     150A      C9 . . .      RET           ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 175
5866	1508	.	.	*****	
5867	1508	.	.	; ENTLCL - ENTER LOCAL MODE *	
5868	1508	.	.	*****	
5869	1508	.	.	ENTLCL EQU \$	
5870	1508	CD	78 11	CALL CKREDIT ;EDIT MODE ENABLED?	
5871	150E	CA	19 15	JZ ENL100 ;NO - GO INTO LOCAL MODE	
5872	1511	3E	08 .	MVI A,REMOTE ;YES - INHIBIT TRANSITION TO	
5873	1513	21	F3 FF	LXI H,MDFLG2 ;LOCAL MODE	
5874	1516	B6	. .	ORA M ;FORCE REMOTE FLAG ON	
5875	1517	77	. .	MOV M,A	
5876	1518	C9	. .	RET ;RETURN	
5877	1519	.	.	;	
5878	1519	.	.	ENL100 EQU \$	
5879	1519	3E	04 .	MVI A,SETLCL ;SET DATACOM FOR LOCAL	
5880	1518	CD	73 13	CALL DCMCTL ;OPERATION	
5881	151E	3E	EF .	MVI A,377Q-REMSET ;CLEAR REMOTE MODE FLAG	
5882	1520	.	.	*****	
5883	1520	.	.	; CLCMFL - CLEAR COMMON FLAGS *	
5884	1520	.	.	*****	
5885	1520	.	.	;	
5886	1520	.	.	; ENTRY: A = 377B-FLAG BIT TO BE CLEARED	
5887	1520	.	.	;	
5888	1520	.	.	; EXIT : A,H,L DESTROYED	
5889	1520	.	.	;	
5890	1520	.	.	CLCMFL EQU \$	
5891	1520	21	F8 FF	LXI H,CMFLGS	
5892	1523	A6	. .	ANA M ;CLEAR THE FLAG BIT	
5893	1524	77	. .	MOV M,A ;STORE THE NEW SETTINGS	
5894	1525	C9	. .	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 176
=====
5896     1526      . . .      ;
5897     1526      . . .      ; ENTREM - ENTER REMOTE MODE
5898     1526      . . .      ;
5899     1526      . . .      ENTREM EQU $
5900     1526     CD 78 11    CALL CKEDIT      ;EDIT MODE ENABLED?
5901     1529     CA 34 15    JZ ENR100        ;NO - GO INTO REMOTE MODE
5902     152C     3E F7 .    MVI A,377Q-REMOTE ;YES - INHIBIT
5903     152E     21 F3 FF    LXI H,MDFLG2     ;TRANSITION TO REMOTE MODE
5904     1531     A6 . .     ANA M            ;FORCE REMOTE FLAG OFF
5905     1532     77 . .     MOV M,A
5906     1533     C9 . .     RET              ;RETURN
5907     1534      . . .      ;
5908     1534      . . .      ENR100 EQU $
5909     1534     97 . .     SUB A            ;CLEAR THE DATA PENDING
5910     1535     32 70 FF    STA MFLGS       ;FLAGS
5911     1538     3E FC .    MVI A,377Q-SBINRY-SDVREC
5912     153A     CD 66 05    CALL CLRMF2     ;CLEAR BINARY RECORD PENDING
5913     153D     3E 03 .    MVI A,SETREM    ;SET DATACOM FOR REMOTE
5914     153F     CD 7E 13    CALL DCMCT1     ;OPERATION
5915     1542     3E 10 .    MVI A,REMSET    ;SET REMOTE MODE FLAG
5916     1544      . . .      ;*****
5917     1544      . . .      ; STCMFL - SET COMMON FLAGS *
5918     1544      . . .      ;*****
5919     1544      . . .      ;
5920     1544      . . .      ; ENTRY: A = FLAG BIT TO BE SET
5921     1544      . . .      ;
5922     1544      . . .      ; EXIT : A,H,L DESTROYED
5923     1544      . . .      ;
5924     1544      . . .      STCMFL EQU $
5925     1544     21 F8 FF    LXI H,CMFLGS
5926     1547     B6 . .     ORA M            ;ADD BIT TO "CMFLGS"
5927     1548     77 . .     MOV M,A          ;STORE NEW SETTINGS
5928     1549     C9 . .     RET              ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 177
5930	154A	.	.	.	;
5931	154A	.	.	.	; * * * * *
5932	154A	.	.	.	;
5933	154A	.	.	.	; FCTKEY - FUNCTION KEY PRESSED (F1-F8)
5934	154A	.	.	.	;
5935	154A	.	.	.	; ENTRY: C = FUNCTION KEY CODE (360-367B)
5936	154A	.	.	.	;*****
5937	154A	.	.	.	; SOFT RETURN = F0, KEY CODE = 357B
5938	154A	.	.	.	;*****
5939	154A	.	.	.	;
5940	154A	.	.	.	; EXIT : DFLGS(FCTK2D) = 1, FUNCTION KEY
5941	154A	.	.	.	; DATA TO BE USED AS NORMAL
5942	154A	.	.	.	; KEYBOARD CHARACTERS
5943	154A	.	.	.	; DFLGS(FCTK2D) = 0
5944	154A	.	.	.	; MFLGS(SFCTKY) = 0, KEY WAS
5945	154A	.	.	.	; INTERPRETED LOCALLY ONLY
5946	154A	.	.	.	; MFLGS(SFCTKY) = 1, DATA WAITING
5947	154A	.	.	.	; FOR BLOCK TRANSFER TRIGGER TO
5948	154A	.	.	.	; SEND TO CPU
5949	154A	.	.	.	; ALL REGISTERS DESTROYED
5950	154A	.	.	.	;
5951	154A	.	.	.	FCTKEY EQU \$
5952	154A	79	.	.	MOV A,C ;COMPUTE NUMBER OF LINES TO
5953	154B	87	.	.	ADD A ;SEARCH:
5954	154C	D6	DD	.	SUI FCTADJ ;2*(FUNCTION NUMBER) - 1
5955	154E	21	A6	FF	LXI H,SFTKYS
5956	1551	CD	FF	0B	CALL MLKSC1 ;LOCATE THE ATTRIBUTE LINE
5957	1554	.	.	.	;
5958	1554	.	.	.	; DEFINITION FOUND - PERFORM FUNCTION
5959	1554	.	.	.	;
5960	1554	7D	.	.	MOV A,L ;COMPUTE LOCATION OF
5961	1555	D6	08	.	SUI ATBLOC ;ATTRIBUTE CODE
5962	1557	SF	.	.	MOV E,A
5963	1558	54	.	.	MOV D,H
5964	1559	CD	C6	1A	CALL CHAIN ;GET ADDRESS OF DATA LINE
5965	155C	22	A4	FF	SHLD CURFKY ;SAVE FIRST CHARACTER ADDRESS
5966	155F	.	.	.	; TO FORCE SKIP OVER "ENDPR"
5967	155F	1A	.	.	LDAX D ;GET ATTRIBUTE CODE
5968	1560	FE	4E	.	CPI N ;NORMAL MODE?
5969	1562	DA	76	15	JC FCT200 ;< - DO LOCAL ONLY
5970	1565	3E	10	.	MVI A,FCTK2D ;(SET DATA XFR FLAG)
5971	1567	CA	6C	18	JZ SETDFL ;YES - SET NORMAL KEY XFR
5972	156A	CD	84	11	CALL GTMODE ;> - SET BLOCK TRANSFER
5973	156D	01	00	20	LXI B,SFCTKY ;FOR FUNCTION KEY
5974	1570	CA	28	18	JZ SBLXFA ;SET FLAG FOR NOT PAGE MODE
5975	1573	C3	30	18	JMP SBLXF1 ;ELSE SET FOR PAGE XFR

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 178
5977	1576	.	.	*****	
5978	1576	.	.	; PERFORM LOCAL ONLY KEY FUNCTION *	
5979	1576	.	.	*****	
5980	1576	.	.	FCT200 EQU \$	
5981	1576	CD	94 16	CALL GTFCTK	;GET FUNCTION KEY DATA
5982	1579	C8	.	RZ	;NONE LEFT - RETURN
5983	157A	.	.	FCT210 EQU \$	
5984	157A	32	9C FF	STA CHARIN	;SAVE FUNCTION KEY CHARACTER
5985	157D	CD	E2 03	CALL CHINT	;PROCESS DATA LOCALLY
5986	1580	3A	9C FF	LDA CHARIN	;RECALL FUNCTION CHARACTER
5987	1583	FE	0D .	CPI CR	;IS IT A RETURN?
5988	1585	C2	76 15	JNZ FCT200	;NO - DO THE NEXT BYTE
5989	1588	3A	F3 FF	LDA MDFLG2	;YES - GET HARD MODE FLAGS
5990	1588	E6	04 .	ANI AUTOLF	;AUTO LINE FEED ENABLED?
5991	158D	CA	76 15	JZ FCT200	;NO - DO NEXT FUNCTION BYTE
5992	1590	0E	0A .	MVI C,LF	;YES - PERFORM LINE FEED
5993	1592	C3	7A 15	JMP FCT210	;FUNCTION


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 180
=====
6024     15B4     . . . ;
6025     15B4     . . . ; MNMDON - MONITOR MODE ON
6026     15B4     . . . ;
6027     15B4     . . . MNMDON EQU $
6028     15B4     3E 08 . MVI A,SETMON ;SET DATACOM MONITOR
6029     15B6     CD 73 13 CALL DCMCTL ;MODE
6030     15B9     C0 . . RNZ ;DON'T MONITOR IF NOT SET
6031     15BA     06 FF . MVI B,377Q ;SET TO BLINK LED
6032     15BC     C3 C1 15 JMP FDO100 ;SET FUNCTION TABLE
6033     15BF     . . . ;*****
6034     15BF     . . . ; FDISON - TURN ON FUNCTION DISABLE MODE *
6035     15BF     . . . ;*****
6036     15BF     . . . FDISON EQU $
6037     15BF     06 00 . MVI B,0 ;SET FOR NO BLINK
6038     15C1     . . . FDO100 EQU $
6039     15C1     3E 01 . MVI A,DSPFNC ;TURN ON DISPLAY FUNCTIONS
6040     15C3     CD 0E 48 CALL ZSTMD1 ;FLAG
6041     15C6     21 43 7F LXI H,FDISTB ;SET H,L TO NEW RANGE TABLE
6042     15C9     C3 5D 05 JMP ESCEN1 ;SET RANGE TABLE AND EXIT
6043     15CC     . . . ;*****
6044     15CC     . . . ; FDISOF - TURN OFF FUNCTION DISABLE *
6045     15CC     . . . ;*****
6046     15CC     . . . FDISOF EQU $
6047     15CC     CD E5 1A CALL CHKSFK ;SOFT KEY DEFINE MODE?
6048     15CF     CA D8 15 JZ FOF010 ;NO - DO NORMAL PROCESSING
6049     15D2     CD D7 13 CALL DCXB2D ;INPUT FROM KEYBOARD?
6050     15D5     C4 96 0D CNZ SFKYOF ;NO - RESTORE NORMAL DISPLAY
6051     15D8     . . . FOF010 EQU $
6052     15D8     CD FA 15 CALL FDESC1 ;DISPLAY INPUT CHARACTER
6053     15D8     3A 69 FF LDA LCHAR
6054     15DE     FE 1B . CPI ESC ;WAS THE LAST CHAR <ESC>?
6055     15E0     C0 . . RNZ ;NO - RETURN
6056     15E1     . . . ; YES - TURN OFF DISPLAY FCTS
6057     15E1     . . . DFCTOF EQU $
6058     15E1     3E 09 . MVI A,SETNRM ;RESTORE DATACOM TO
6059     15E3     CD 11 50 CALL ZDCCTL ;NORMAL MODE
6060     15E6     CD 48 05 CALL ESCEND ;YES - TURN OFF DISABLE MODE
6061     15E9     3E 01 . MVI A,DSPFNC ;TURN OFF DISPLAY FUNCTIONS
6062     15EB     C3 11 48 JMP ZCLMD1 ;FLAG
6063     15EE     . . . ;*****
6064     15EE     . . . ; FUNCTION DISABLE ESCAPE *
6065     15EE     . . . ;*****
6066     15EE     . . . FDESC EQU $
6067     15EE     CD E5 1A CALL CHKSFK ;SOFT KEY DEFINE MODE?
6068     15F1     CA FA 15 JZ FDESC1 ;NO - DO NORMAL PROCESSING
6069     15F4     CD D7 13 CALL DCXB2D ;INPUT FROM KEYBOARD?
6070     15F7     C4 96 0D CNZ SFKYOF ;NO - RESTORE NORMAL DISPLAY
6071     15FA     . . . FDESC1 EQU $
6072     15FA     CD 83 25 CALL DSPCHR ;DISPLAY THE ESCAPE CODE
6073     15FD     C3 AF 21 JMP CRADV1 ;RESET CURSOR ADVANCE FLAG T
=====

```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 182
=====
6077     1600      . . .      ;*****
6078     1600      . . .      ; FUNCTION TABLE FOR TERMINAL FUNCTION KEYS *
6079     1600      . . .      ;*****
6080     1600      . . .      FNCTAB EQU $
6081     1600      09 14      DW  DPSEND      ;230 - ENTER KEY
6082     1602      6C 13      DW  BRKDC       ;231 - BREAK KEY
6083     1604      E1 15      DW  DFCTOF      ;232 - DISPLAY FUNCTIONS OFF
6084     1606      DE 16      DW  IOKEYS      ;233 - I/O CONTROL KEY
6085     1608      DE 16      DW  IOKEYS      ;234 - READ KEY
6086     160A      DE 16      DW  IOKEYS      ;235 - RECORD KEY
6087     160C      DE 16      DW  IOKEYS      ;236 - SELECT KEY
6088     160E      DE 16      DW  IOKEYS      ;237 - CONDITION TAPES
6089     1610      DE 16      DW  IOKEYS      ;240 - (CONTROL) READ KEY
6090     1612      . . .      ;
6091     0098      . . .      ENTRCD EQU 230Q ;ENTER KEY CODE
6092     009D      . . .      RCKYCD EQU 235Q ;RECORD KEY CODE
6093     009E      . . .      SLKYCD EQU 236Q ;SELECT KEY CODE
6094     00A0      . . .      CTRDKY EQU 240Q ;CONTROL READ KEY CODE
6095     0098      . . .      FNCLWR EQU 230Q ;FUNCTION CODE LOWER LIMIT
6096     00A1      . . .      FNCLIM EQU 241Q ;FUNCTION CODE UPPER LIMIT
6097     1612      . . .      ;
6098     008E      . . .      ESCSO EQU 216Q  ;<ESC>-<SO> CODE
6099     00E4      . . .      ESCLWD EQU 344Q ;<ESC>-<LOWER CASE D> CODE
6100     1612      . . .      ;*****
6101     00EF      . . .      F0CODE EQU 357Q ;CODE FOR F0 = SOFT RET KEY
6102     1612      . . .      ;*****
6103     00F0      . . .      F1CODE EQU 360Q ;F1 CODE
6104     1612      . . .      ;*****
6105     01DE      . . .      FCTAD1 EQU F0CODE*2 ;FUNCTION CODE ADJUSTMENT
6106     1612      . . .      ;*****
6107     FFDD      . . .      FCTADJ EQU -FCTAD1/256*256+FCTAD1-1 ;FACTOR
6108     00FD      . . .      STFUR2 EQU 375Q ;SET FOREIGN MODE STEP 2
6109     00FE      . . .      STFUR1 EQU 376Q ;SET FOREIGN MODE STEP 1
6110     00FF      . . .      ENHNCF EQU 377Q ;ENHANCE DISPLAY FUNCTION
6111     1612      . . .      ;*****
6112     1612      . . .      ; FUNCTION ADDRESSES FOR I/O KEYS *
6113     1612      . . .      ;*****
6114     1612      . . .      IOKYTB EQU $
6115     1612      02 28      DW  IOKEY      ;I/O CONTROL KEY
6116     1614      05 28      DW  REDKEY     ;READ KEY
6117     1616      0B 28      DW  REKEY      ;RECORD KEY
6118     1618      0E 28      DW  SELKEY     ;SELECT KEY
6119     161A      14 28      DW  CONDTN     ;CONDITION TAPES
6120     161C      08 28      DW  CTLRED     ;(CONTROL) READ KEY
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 183
=====
6122     161E      . . .      ;
6123     161E      . . .      ; DISPLAY STRINGS FOR SOFT KEY DISPLAY
6124     161E      . . .      ;
6125     161E      . . .      ATBLIN EQU $
6126     161E      CC 20 1B    DB EOL,ABLNK,ESC,ENDPR
6127     1622      . . .      ;
6128     1622      CC C0 .     DB EOL,STPR
6129     1624      . . .      ATB010 EQU $
6130     1624      54 C8 C1    DB 'T',SFKYAT,ENDPR,' '
6131     1628      80 30 66    DB NORMAL,'0',146Q,INVRS,0
6132     0008      . . .      ATBLOC EQU $-ATB010-1 ;ATTRIBUTE LOCATION IN BLK
6133     000E      . . .      ATBLEN EQU $-ATBLIN-1 ;ATTRIBUTE LINE LENGTH
6134     0002      . . .      CHRLOC EQU 2 ;CHARACTER LOCATION IN STRIN
6135     162D      . . .      ;***** GRAPHICS MODIFICATION *****
6136     162D      . . .      RTNKEY EQU $
6137     162D      82 43 52    DB INVRS,'CR',NORMAL,' '
6138     1632      C1 C8 4E    DB ENDPR,SFKYAT,'N',STPR,EOL
6139     1637      C1 0D CC    DB ENDPR,CR,EOL,0
6140     163B      . . .      ;*****
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 184
=====
6142     163B      . . .      ;*****
6143     163B      . . .      ; FNDTAB - FIND TAB MASK *
6144     163B      . . .      ; EXIT L,H = ADDR OF BYTE CONTAINING TAB BIT *
6145     163B      . . .      ;      A = MASK FOR TAB BIT *
6146     163B      . . .      ;*****
6147     163B      . . .      FNDTAB EQU $
6148     163B      3A C1 FF      LDA CURCOL      ;GET CURSOR COLUMN
6149     163E      47 . .      MOV B,A         ;SAVE IN B
6150     163F      . . .      FNDTB1 EQU $
6151     163F      E6 F8 .      ANI 370Q        ;MASK OFF 3 LSB'S
6152     1641      0F . .      RRC             ;RIGHT-ADJUST MSB'S
6153     1642      0F . .      RRC
6154     1643      0F . .      RRC
6155     1644      C6 78 .      ADI HTBTBL     ;ADD BASE OF TAB TABLE
6156     1646      6F . .      MOV L,A        ;SAVE IN L
6157     1647      78 . .      MOV A,B        ;GET CURSOR COLUMN
6158     1648      E6 07 .      ANI 7          ;GET 3 LSB'S
6159     164A      47 . .      MOV B,A        ;SAVE IN B
6160     164B      04 . .      INR B          ;ADJUST BIT NUMBER
6161     164C      . . .      ;*****
6162     164C      . . .      ; FNDTB2 - SET BIT N *
6163     164C      . . .      ;*****
6164     164C      . . .      ;
6165     164C      . . .      ; ENTRY:  B = BIT NUMBER TO BE SET
6166     164C      . . .      ;
6167     164C      . . .      ; EXIT :  A = BYTE WITH BIT N SET
6168     164C      . . .      ;      B = 0
6169     164C      . . .      ;
6170     164C      . . .      FNDTB2 EQU $
6171     164C      3E 80 .      MVI A,200Q     ;SET BIT 7 OF A
6172     164E      . . .      FTB100 EQU $
6173     164E      07 . .      RLC            ;SHIFT LEFT 1 POSITION
6174     164F      05 . .      DCR B          ;DECREMENT BIT COUNT
6175     1650      C2 4E 16     JNZ FTB100     ;CONTINUE IF NOT DONE
6176     1653      C9 . .      RET            ;RETURN
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 185
=====
6178     1654      . . .      ;*****
6179     1654      . . .      ; EXIT FORMAT MODE *
6180     1654      . . .      ;*****
6181     1654      . . .      FORMOF EQU $
6182     1654     3E 08 .      MVI  A,FORMAT ;SET BIT TO BE CLEARED
6183     1656     32 C2 FF      STA  PROFLD  ;SET PROTECT FLAG FOR UNPROT
6184     1659     C3 11 48      JMP  ZCLMD1  ;CLEAR FORMAT MODE FLAG
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 186
6186	165C	.	.	;*****	
6187	165C	.	.	; FRECNT - CHECK THE NUMBER OF FREE BLOCKS *	
6188	165C	.	.	;	
6189	165C	.	.	; EXIT: Z=F, NOT ENOUGH FREE BLOCKS *	
6190	165C	.	.	; Z=T, ENOUGH FREE BLOCKS *	
6191	165C	.	.	;*****	
6192	165C	.	.	FRCNT EQU \$	
6193	165C	06	19	MVI B,25 ;SET DESIRED NUMBER OF BLOCK	
6194	165E	11	AA FF	LXI D,FRBLKS-2 ;SET TO FREE LIST HEAD	
6195	1661	.	.	FRC010 EQU \$	
6196	1661	EB	.	XCHG ;SET H,L TO ADDRESS OF LSB	
6197	1662	23	.	INX H ;PART OF PREVIOUS LINE	
6198	1663	23	.	INX H ;POINTER	
6199	1664	7E	.	MOV A,M ;GET LSB OF NEXT LINE LINK	
6200	1665	B7	.	ORA A ;ANY MORE FREE BLOCKS?	
6201	1666	CA	82 16	JZ FRC100 ;NO - TRY TO GET MORE	
6202	1669	05	.	DCR B ;ENOUGH FREE BLOCKS?	
6203	166A	C8	.	RZ ;YES - RETURN SUCCESSFUL	
6204	166B	CD	C6 1A	CALL CHAIN ;NO - GET NEXT LINE ADDRESS	
6205	166E	54	.	MOV D,H ;SAVE NEXT LINE ADDRESS IN	
6206	166F	5D	.	MOV E,L ;D,E	
6207	1670	.	.	FRC050 EQU \$	
6208	1670	E6	F0	ANI 360Q ;COMPUTE ADDRESS OF NEXT	
6209	1672	6F	.	MOV L,A ;BLOCK LINK	
6210	1673	7E	.	MOV A,M ;GET THE LSB OF THE LINK	
6211	1674	2F	.	CMA ;A IS IT AN EOL LINK (LOWER	
6212	1675	E6	0F	ANI BLKSM ;FOUR BITS NOT ALL UNES)?	
6213	1677	C2	61 16	JNZ FRC010 ;NO - GO TO THE NEXT LINE	
6214	167A	05	.	DCR B ;ENOUGH FREE BLOCKS FOUND?	
6215	167B	C8	.	RZ ;YES - RETURN SUCCESSFUL	
6216	167C	CD	C6 1A	CALL CHAIN ;NO - GO TO THE NEXT BLOCK	
6217	167F	C3	70 16	JMP FRC050 ;CHECK FOR END OF LINE	
6218	1682	.	.	;*****	
6219	1682	.	.	; NOT ENOUGH FREE BLOCKS - TRY TO GET MORE *	
6220	1682	.	.	;*****	
6221	1682	.	.	FRC100 EQU \$	
6222	1682	CD	10 07	CALL PTBLK ;REMOVE A LINE FROM DISPLAY	
6223	1685	C2	5C 16	JNZ FRECNT ;RECOUNT IF LINE FREED	
6224	1688	3C	.	INR A ;(FORCE NZ)	
6225	1689	C9	.	RET ;RETURN FAIL OTHERWISE	


```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 187
=====
6227     168A      . . .      ;*****
6228     168A      . . .      ; FRNCT1 - FOREIGN MODE CONTROL 1 (<ESC>-"<") *
6229     168A      . . .      ;*****
6230     168A      . . .      FRNCT1 EQU $
6231     168A      3E 0E .      MVI A,FRNMD1 ;SET KEYBOARD FOREIGN MODE 1
6232     168C      C3 08 48     JMP ZKBCTL
6233     168F      . . .      ;*****
6234     168F      . . .      ; FRNCT2 - FOREIGN MODE CONTROL 2 (<ESC>-">") *
6235     168F      . . .      ;*****
6236     168F      . . .      FRNCT2 EQU $
6237     168F      3E 0F .      MVI A,FRNMD2 ;SET KEYBOARD FOREIGN MODE 2
6238     1691      C3 08 48     JMP ZKBCTL
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 188
=====
6240     1694      . . .      ;
6241     1694      . . .      ; * * * * *
6242     1694      . . .      ;
6243     1694      . . .      ;   GTFCTK - GET FUNCTION KEY
6244     1694      . . .      ;
6245     1694      . . .      ;       ENTRY:  DON'T CARE
6246     1694      . . .      ;
6247     1694      . . .      ;       EXIT :  NZ - FUNCTION KEY CHAR AVAILABLE
6248     1694      . . .      ;                A = C = FUNCTION KEY CHARACTER
6249     1694      . . .      ;                Z - NO FUNCTION KEY CHAR AVAILABLE
6250     1694      . . .      ;                DFLGS(FCTK2D) = 0
6251     1694      . . .      ;                A DESTROYED
6252     1694      . . .      ;                D-L DESTROYED
6253     1694      . . .      ;
6254     1694      . . .      ;   GTFCTK EQU $
6255     1694      2A  A4  FF      ;   LHLD CURFKY      ;GET LAST FUNCTION KEY
6256     1697      . . .      ;                        CHARACTER ADDRESS
6257     1697      CD  8F  0C      ;   CALL NXTCHO     ;GET THE NEXT CHARACTER
6258     169A      C2  A5  16      ;   JNZ  GTF010     ;EOL LINK - DO EOL EXIT
6259     169D      EB  . .      ;   XCHG
6260     169E      22  A4  FF      ;   SHLD CURFKY     ;STORE NEW ADDRESS
6261     16A1      FE  80  .      ;   CPI  ADEL+1     ;IS CHARACTER ASCII?
6262     16A3      4F  . .      ;   MOV  C,A        ;(PUT DATA IN C-REGISTER)
6263     16A4      F8  . .      ;   RM              ;YES - RETURN
6264     16A5      . . .      ;
6265     16A5      . . .      ;   EOL FOUND - CLEAR FCTK2D FLAG
6266     16A5      . . .      ;
6267     16A5      . . .      ;   GTF010 EQU $
6268     16A5      3E  EF  .      ;   MVI  A,377Q-FCTK2D
6269     16A7      CD  53  17      ;   CALL CLRDFL     ;CLEAR FLAG FROM FLAG WORD
6270     16AA      BF  . .      ;   CMP  A          ;SET Z TRUE
6271     16AB      C9  . .      ;   RET            ;RETURN
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 189
=====
6273     16AC      . . .      ;*****
6274     16AC      . . .      ; HTBSET - TAB SET ROUTINE *
6275     16AC      . . .      ;*****
6276     16AC      . . .      HTBSET EQU $
6277     16AC      CD 3B 16    CALL FNDTAB ;GET TABLE ENTRY FOR COLUMN
6278     16AF      B6 . .     ORA M ;SET TAB
6279     16B0      77 . .     MOV M,A
6280     16B1      C9 . .     RET ;RETURN
6281     16B2      . . .      ;*****
6282     16B2      . . .      ; HTBCLR - TAB CLEAR ROUTINE *
6283     16B2      . . .      ;*****
6284     16B2      . . .      HTBCLR EQU $
6285     16B2      CD 3B 16    CALL FNDTAB ;GET TABLE ENTRY FOR COLUMN
6286     16B5      EE FF .    XRI 3770 ;COMPLEMENT MASK
6287     16B7      A6 . .     ANA M ;CLEAR TAB
6288     16B8      77 . .     MOV M,A
6289     16B9      C9 . .     RET ;RETURN
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 190
=====
6291     16BA      . . .      ;*****
6292     16BA      . . .      ; IOBNGO - FAST BINARY READ ESCAPE SEQUENCE *
6293     16BA      . . .      ;*****
6294     16BA      . . .      IOBNGO EQU $
6295     16BA      21 2C 28      LXI H,CTDCDP ;EXECUTE FAST BINARY READ
6296     16BD      C3 E5 16      JMP IORMGO ;IF I/O ROM PRESENT
6297     16C0      . . .      ;*****
6298     16C0      . . .      ; IOBSYC - WAIT FOR CTU IDLE *
6299     16C0      . . .      ;*****
6300     16C0      . . .      IOBSYC EQU $
6301     16C0      21 3A 28      LXI H,BSYCHK ;GO TO CTU BUSY CHECK
6302     16C3      CD E5 16      CALL IORMGO ;ROUTINE
6303     16C6      3A 55 FF      LDA CMND ;GET CURRENT CTU COMMAND
6304     16C9      E6 01 .      ANI RUN ;TAPE STILL RUNNING?
6305     16CB      C8 . .      RZ ;NO - RETURN
6306     16CC      32 4F FF      STA IOCERR ;YES - CLEAR "IOCERR"
6307     16CF      C3 C0 16      JMP IOBSYC ;CONTINUE WAITING
6308     16D2      . . .      ;*****
6309     16D2      . . .      ; IOCTGO - I/O CONTROL ESCAPE SEQUENCE *
6310     16D2      . . .      ;*****
6311     16D2      . . .      IOCTGO EQU $
6312     16D2      21 1A 28      LXI H,IOCNTL ;EXECUTE I/O CONTROL ESCAPE
6313     16D5      C3 E5 16      JMP IORMGO ;SEQ IF I/O ROM PRESENT
6314     16D8      . . .      ;*****
6315     16D8      . . .      ; IOCTMN - MONITOR CARTRIDGE TAPES *
6316     16D8      . . .      ;*****
6317     16D8      . . .      IOCTMN EQU $
6318     16D8      21 2F 28      LXI H,CTMON ;GET MONITOR ADDRESS
6319     16D8      C3 E5 16      JMP IORMGO ;EXECUTE IF CODE PRESENT
6320     16DE      . . .      ;*****
6321     16DE      . . .      ; IOKEYS - I/O KEY HIT *
6322     16DE      . . .      ;*****
6323     16DE      . . .      ;
6324     16DE      . . .      ; ENTRY: D,E = KEY INDEX
6325     16DE      . . .      ;
6326     16DE      . . .      IOKEYS EQU $
6327     16DE      21 0C 16      LXI H,IOKYTB-6
6328     16E1      19 . .      DAD D ;COMPUTE KEY FUNCTION ADDRESS
6329     16E2      CD C6 1A      CALL CHAIN ;EXECUTE KEY FUNCTION IF I/O
6330     16E5      . . .      ; ROM PRESENT
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 191
6332	16E5	. . .	;*****	
6333	16E5	. . .	; IORMGO - PERFORM FUNCTION IF OPTION ROMS *	
6334	16E5	. . .	; ARE PRESENT *	
6335	16E5	. . .	;*****	
6336	16E5	. . .	;	
6337	16E5	. . .	; ENTRY: H,L = VECTOR TO BE ENTERED	
6338	16E5	. . .	;	
6339	16E5	. . .	; EXIT : NC - FUNCTION EXECUTED	
6340	16E5	. . .	; REGISTERS SET ACCORDING TO FUNCTION	
6341	16E5	. . .	; C - FUNCTION NOT EXECUTED	
6342	16E5	. . .	; A DESTROYED	
6343	16E5	. . .	;	
6344	16E5	. . .	IORMGO EQU \$	
6345	16E5	E5 . .	PUSH H ;PUT FUNCTION ADDR ON STACK	
6346	16E6	2E 00 .	MVI L,0 ;CHECK ROM START LOCATION	
6347	16E8	CD F5 16	CALL IORMG1 ;DOES ROM EXIST?	
6348	16EB	C8 . .	RZ ;YES - EXECUTE FUNCTION	
6349	16EC	21 83 10	LXI H,NODRVR ;NO - SET ERROR MESSAGE TO	
6350	16EF	22 F1 FF	SHLD MSGPT1 ;"NO DEVICE DRIVER"	
6351	16F2	E1 . .	POP H ;RESTORE STACK	
6352	16F3	37 . .	STC ;RETURN FUNCTION NOT	
6353	16F4	C9 . .	RET ;EXECUTED (C-TRUE)	
6354	16F5	. . .	;*****	
6355	16F5	. . .	; IORMG1 - CHECK FOR PRESENCE OF OPTION ROM *	
6356	16F5	. . .	;*****	
6357	16F5	. . .	;	
6358	16F5	. . .	; ENTRY: H,L = ROM STARTING ADDRESS	
6359	16F5	. . .	;	
6360	16F5	. . .	; EXIT : Z => ROM EXIST	
6361	16F5	. . .	; H,L = H,L(ENTRY)+1	
6362	16F5	. . .	; NZ => ROM ABSENT	
6363	16F5	. . .	; A DESTROYED	
6364	16F5	. . .	; H,L = H,L(ENTRY) => ROM ABSENT	
6365	16F5	. . .	; H,L = H,L(ENTRY)+1 => WRONG ROM	
6366	16F5	. . .	;	
6367	16F5	. . .	IORMG1 EQU \$	
6368	16F5	7E . .	MOV A,M ;GET FIRST ROM BYTE	
6369	16F6	E6 F0 .	ANI 3600 ;CHECK UPPER 4 BITS ONLY	
6370	16F8	FE 50 .	CPI P ;IS IT AN UPPER CASE P?	
6371	16FA	C0 . .	RNZ ;NO - RETURN ROM ABSENT	
6372	16FB	23 . .	INX H ;YES - CHECK SECOND BYTE	
6373	16FC	7E . .	MOV A,M ;SECOND BYTE OF ROM SHOULD	
6374	16FD	BC . .	CMP H ;EQUAL HIGH ORDER EIGHT	
6375	16FE	C9 . .	RET ;BITS IN ITS PROPER ADDRES	
6376	16FF	. . .	; RANGE	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 192
6378	16FF	.	.	*****	
6379	16FF	.	.	; IOINTR - I/O INTERRUPT PROCESSING *	
6380	16FF	.	.	*****	
6381	16FF	.	.	;	
6382	16FF	.	.	; ENTRY: "PSW" AND H,L PUSHED	
6383	16FF	.	.	; A = INTERRUPT CODE	
6384	16FF	.	.	;	
6385	16FF	.	.	IOINTR EQU \$	
6386	16FF	CD	65 91	CALL INTVEC ;CHECK ALTERNATE INTERRUPT	
6387	1702	3A	FS FF	LDA PRCCTL ;GET CURRENT PROCESSOR STATE	
6388	1705	F6	40 .	ORI POLL ;POLL THE I/O BOARDS TO FIND	
6389	1707	D3	70 .	OUT PROCSR ;OUT WHO INTERRUPTED	
6390	1709	21	00 87	LXI H,IOCRCL ;DO DUMMY I/O READ TO GET	
6391	170C	6E	. .	MOV L,M ;POLL RESPONSE	
6392	170D	E6	BF .	ANI 377Q-POLL	
6393	170F	D3	70 .	OUT PROCSR ;RESTORE PROCESSOR STATE	
6394	1711	3A	7F FE	LDA DEVFLG ;GET POLL DEVICE FLAG	
6395	1714	A5	. .	ANA L ;DEVICE DRIVER PRESENT?	
6396	1715	FA	3D 28	JM CTINTR ;CTU - DO CTU ROUTINE	
6397	1718	87	. .	ADD A ;ALTERNATE I/O INTERRUPT?	
6398	1719	FA	08 92	JM ZINTAL ;YES - GO CHECK INTERRUPT	
6399	171C	.	.	*****	
6400	171C	.	.	; INVALID DEVICE INTERRUPT - REPORT ERROR *	
6401	171C	.	.	*****	
6402	171C	7D	. .	MOV A,L ;RECALL POLL RESPONSE	
6403	171D	06	40 .	MVI B,ATSIGN ;COMPUTE ERROR CODE	
6404	171F	B7	. .	ORA A ;ANY DEVICE INTERRUPTED?	
6405	1720	CA	28 17	JZ IOI020 ;NO - DON'T LOOK FOR BIT	
6406	1723	.	.	; YES - DETERMINE DEVICE	
6407	1723	.	.	IOI010 EQU \$	
6408	1723	04	. .	INR B ;INCREMENT ERROR CODE	
6409	1724	07	. .	RLC ;DEVICE TYPE FOUND?	
6410	1725	D2	23 17	JNC IOI010 ;NO - CONTINUE LOOKING	
6411	1728	.	.	IOI020 EQU \$;YES - SET ERROR CODE	
6412	1728	78	. .	MOV A,B	
6413	1729	.	.	; FALL INTO ERROR REPORTER	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 193
=====
6415     1729      . . .      ;*****
6416     1729      . . .      ; INTERR - REPORT INVALID INTERRUPT OCCURRED *
6417     1729      . . .      ;*****
6418     1729      . . .      ;
6419     1729      . . .      ; ENTRY:  A = ERROR CODE (ASCII CHARACTER)
6420     1729      . . .      ;
6421     1729      . . .      INTERR EQU $
6422     1729     21 DE FF      LXI  H,IODATA  ;SET ERROR CODE FOR ERROR
6423     172C     22 EF FF      SHLD MSGPT2    ;MESSAGE
6424     172F     77 . .      MOV  M,A
6425     1730     23 . .      INX  H
6426     1731     36 CE .      MVI  M,EOP
6427     1733     21 69 10     LXI  H,INERMS  ;REPORT INTERRUPT ERROR
6428     1736     AF . .      XRA  A         ;STOP ANY CTU MOTION
6429     1737     32 00 8B     STA  IOCTCO
6430     173A     C3 85 13     JMP  HANGUO    ;AND HANG TERMINAL
6431     173D      . . .      ;*****
6432     173D      . . .      ; INTRPT - PROCESS UNEXPECTED INTERRUPTS *
6433     173D      . . .      ;*****
6434     173D      . . .      ;
6435     173D      . . .      ; ENTRY:  "PSW" PUSHED
6436     173D      . . .      ;         A = INTERRUPT CODE
6437     173D      . . .      ;         C-FLAG CLEARED
6438     173D      . . .      ;
6439     173D      . . .      INTRPT EQU $
6440     173D     CD 65 91     CALL INTVEC    ;ANY INTERRUPT HANDLER?
6441     1740     D2 29 17     JNC  INTERR    ;NO - REPORT ERROR
6442     1743     F1 . .      POP  PSW       ;YES - RESTORE PSW
6443     1744     FB . .      EI            ;RE-ENABLE INTERRUPTS
6444     1745     C9 . .      RET           ;RETURN TO INTERRUPTED CODE
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 194
=====
6446     1746      .      .      .      ;*****
6447     1746      .      .      .      ; IOINTR - I/O INTERRUPT PROCESSING *
6448     1746      .      .      .      ;*****
6449     1746      .      .      .      ;*****
6450     1746      .      .      .      ; KEYBOARD ENABLE *
6451     1746      .      .      .      ;*****
6452     1746      .      .      .      KBEN      EQU $
6453     1746      3A     6E     FF      LDA      DFLGS
6454     1749      E6     40      .      ANI      KBDLOK      ;KEYBOARD LOCKED BY ESC SEQ?
6455     1748      C0      .      .      RNZ      ;YES - DO NOT UNLOCK KEYBOARD
6456     174C      .      .      .      ;
6457     174C      .      .      .      KBEN1     EQU $
6458     174C      3E     02      .      MVI     A,UNLKKB      ;UNLOCK THE KEYBOARD
6459     174E      CD     08     48      CALL    ZKBCTL
6460     1751      3E     BF      .      MVI     A,3770-KBDLOK ;CLEAR LOCKED FLAG
6461     1753      .      .      .      ;
6462     1753      .      .      .      ; CLRDFL - CLEAR DATA TRANSFER FLAGS
6463     1753      .      .      .      ;
6464     1753      .      .      .      ; ENTRY:  A = FLAGS TO BE CLEARED
6465     1753      .      .      .      ;
6466     1753      .      .      .      CLRDFL   EQU $
6467     1753      21     6E     FF      LXI     H,DFLGS
6468     1756      A6      .      .      ANA     M              ;MASK OUT FLAGS
6469     1757      .      .      .      ;
6470     1757      .      .      .      ; STOREA - STORE VALUE N A-REG AND RETURN
6471     1757      .      .      .      ;
6472     1757      .      .      .      ; ENTRY:  A = VALUE TO BE STORED
6473     1757      .      .      .      ; H,L = LOCATION TO BE STORED IN
6474     1757      .      .      .      ;
6475     1757      .      .      .      STOREA   EQU $
6476     1757      77      .      .      MOV     M,A            ;STORE UPDATED VALUE
6477     1758      C9      .      .      RET      ;RETURN
6478     1759      .      .      .      ;*****
6479     1759      .      .      .      ; KEYBOARD LOCK *
6480     1759      .      .      .      ;*****
6481     1759      .      .      .      KBLOK0   EQU $
6482     1759      3E     40      .      MVI     A,KBDLOK      ;SET ESCAPE SEQUENCE LOCK
6483     175B      CD     6C     18      CALL    SETDFL        ;FLAG
6484     175E      .      .      .      ;
6485     175E      .      .      .      KBLOK    EQU $
6486     175E      3E     01      .      MVI     A,LOCKKB      ;LOCK THE KEYBOARD
6487     1760      C3     08     48      JMP     ZKBCTL
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 195
6489	1763	.	.	*****	
6490	1763	.	.	; ESC & LOWER CASE B	*
6491	1763	.	.	; BINARY LOADER	*
6492	1763	.	.	; A SET ADDRESS = DIGITS	*
6493	1763	.	.	; C COMPARE CHECKSUM	*
6494	1763	.	.	; D STORE BYTE	*
6495	1763	.	.	; INCREMENT ADDRESS	*
6496	1763	.	.	; E CALL ADDRESS	*
6497	1763	.	.	; DIGITS 1,2,3,4, OR 5	*
6498	1763	.	.	*****	
6499	1763	.	.	LOADR EQU \$;INITIAL ENTRY	
6500	1763	3E	18	MVI A,MAXROW+1	
6501	1765	32	C0	STA CURROW ;SET CURSOR OFF THE SCREEN	
6502	1768	21	74	LXI H,LDRMSG	
6503	1768	CD	2F	CALL DSPMS0 ;DISPLAY THE LOADER MESSAGE	
6504	176E	.	.	LOADR1 EQU \$;ENTRY TO NOT DISPLAY MESSAG	
6505	176E	21	00	LXI H,0 ;CLEAR CHECKSUM ACCUMULATOR	
6506	1771	22	D7	SHLD LCHKSM	
6507	1774	3E	04	MVI A,FRCRST ;SET FORCE RESET FLAG	
6508	1776	CD	44	CALL STCMFL	
6509	1779	.	.	LDR0 EQU \$	
6510	1779	3A	88	LDA CHAR ;RECALL INPUT CHARACTER	
6511	177C	E6	20	ANI 40Q ;IS IT UPPER CASE?	
6512	177E	3E	FB	MVI A,377Q-FRCRST	
6513	1780	CA	20	JZ CLCMFL ;YES - CLEAR FORCE RESET AND	
6514	1783	.	.	; EXIT ESCAPE SEQUENCE	
6515	1783	21	53	LXI H,LDRTAB ;NO - SET LOADER FUNCTION	
6516	1786	3E	08	MVI A,OCTRDY ;SET FOR OCTAL RADIX	
6517	1788	C3	34	JMP ESCAPO	
6518	178B	.	.	;	
6519	178B	.	.	; <A> - ADDRESS PARAMETER - SET ADDRESS	
6520	178B	.	.	;	
6521	178B	.	.	LDR3 EQU \$	
6522	178B	2A	DE	LHLD LDATA ;GET ACCUMULATED DATA	
6523	178E	22	D5	SHLD LADDR ;SET AS LOAD ADDRESS	
6524	1791	EB	.	XCHG ;PUT VALUE INTO D,E	
6525	1792	.	.	LDR035 EQU \$	
6526	1792	2A	D7	LHLD LCHKSM ;ACCUMULATE CHECKSUM	
6527	1795	19	.	DAD D	
6528	1796	22	D7	SHLD LCHKSM	
6529	1799	C3	79	JMP LDR0 ;RETURN TO SYSTEM	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 196
6531	179C	.	.	;	
6532	179C	.	.	;	<D> - DATA BYTE PARAMETER - STORE DATA BYTE
6533	179C	.	.	;	
6534	179C	.	.	LDR4 EQU \$	
6535	179C	2E	DE	MVI L,LDATA-BASE	
6536	179E	5E	.	MOV E,M	;GET ACCUMULATED DATA
6537	179F	2A	D5	LHLD LADDR	;GET LOAD ADDRESS
6538	17A2	73	.	MOV M,E	;STORE THE BYTE
6539	17A3	16	00	MVI D,0	;ZERO MSB FOR CHECKSUM
6540	17A5	23	.	INX H	;INCREMENT AND STORE NEW
6541	17A6	22	D5	SHLD LADDR	;LOAD ADDRESS
6542	17A9	C3	92	JMP LDR035	;ACCUMULATE CHECKSUM
6543	17AC	.	.	;	*****
6544	17AC	.	.	;	<E> - EXECUTE ENTERED CODE, WAIT UNTIL CTU'S *
6545	17AC	.	.	;	STOPPED BEFORE EXECUTING CODE *
6546	17AC	.	.	;	*****
6547	17AC	.	.	LDR060 EQU \$	
6548	17AC	CD	DA	CALL DISLN4	;RE-ENABLE RESET KEY
6549	17AF	CD	17	CALL ZGETDC	;PURGE DATA COMM INPUT
6550	17B2	DC	82	CC DCERR	;PROCESS ERROR IF ANY
6551	17B5	3A	55	LDA CMND	;GET CTU COMMAND
6552	17B8	E6	01	ANI RUN	;CTU'S RUNNING?
6553	17BA	C2	AC	JNZ LDR060	;YES - CONTINUE WAITING
6554	17BD	3E	80	MVI A,CRTOFF	;NO - TURN OFF THE DISPLAY
6555	17BF	32	20	STA IOCRRW	
6556	17C2	F3	.	DI	;DISABLE INTERRUPTS
6557	17C3	2A	D5	LHLD LADDR	;GET LOAD ADDRESS
6558	17C6	E9	.	PCHL	;START EXECUTION THERE
6559	17C7	.	.	;	
6560	17C7	.	.	;	<C> - CHECKSUM ENTRY
6561	17C7	.	.	;	
6562	17C7	.	.	LDR10 EQU \$;CHECKSUM ENTRY
6563	17C7	21	F7	LXI H,ERRFLG	;DEFAULT TO GOOD CHECKSUM
6564	17CA	7E	.	MOV A,M	
6565	17CB	F6	04	ORI LDRCHK	
6566	17CD	77	.	MOV M,A	;SET ERROR FLAGS
6567	17CE	2A	DE	LHLD LDATA	;GET USER SPECIFIED CHECKSUM
6568	17D1	EB	.	XCHG	
6569	17D2	21	D7	LXI H,LCHKSM	
6570	17D5	7B	.	MOV A,E	;COMPARE TO CALCULATED
6571	17D6	AE	.	XRA M	;CHECKSUM
6572	17D7	4F	.	MOV C,A	
6573	17D8	23	.	INX H	
6574	17D9	7A	.	MOV A,D	
6575	17DA	AE	.	XRA M	
6576	17DB	B1	.	URA C	;DO CHECKSUMS MATCH?
6577	17DC	CA	79	JZ LDR0	;YES - RETURN NORMAL
6578	17DF	C7	.	RST RESET	;NO - RESET TERMINAL

13255

2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 197
=====
6580     17E0      . . .      ;*****
6581     17E0      . . .      ; PARAMETERIZED SEQUENCES INITIAL CONTROL *
6582     17E0      . . .      ;*****
6583     17E0      . . .      PRMSEQ EQU $
6584     17E0      21 E9 7E      LXI H,PRMTAB ;SET RANGE TABLE FOR
6585     17E3      C3 32 05      JMP ESCAPA   ;PARAMETERIZED ESC SEQUENC
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 198
=====
6587     17E6     . . .      ;*****
6588     17E6     . . .      ; START PROTECT *
6589     17E6     . . .      ;*****
6590     17E6     . . .      PRSTRT EQU $
6591     17E6     06 C0     .      MVI B,STPR      ;STORE START PROTECT CONTROL
6592     17E8     C3 14 18     .      JMP PRO100      ;FLAG
6593     17EB     . . .      ;*****
6594     17EB     . . .      ; TRANSMIT-ONLY *
6595     17EB     . . .      ;*****
6596     17EB     . . .      STRXMO EQU $
6597     17EB     3E C2     .      MVI A,XMONLY   ;STORE TRANSMIT-ONLY CONTROL
6598     17ED     C3 F2 17     .      JMP PRO010      ;FLAG
6599     17F0     . . .      ;*****
6600     17F0     . . .      ; END PROTECT *
6601     17F0     . . .      ;*****
6602     17F0     . . .      PREND EQU $
6603     17F0     3E C1     .      MVI A,ENDPR    ;STORE END PROTECT CONTROL
6604     17F2     . . .      ;
6605     17F2     . . .      ; MAKE SURE PREVIOUS CHAR IS DEFINED PROTECTED
6606     17F2     . . .      ;
6607     17F2     . . .      PRO010 EQU $
6608     17F2     32 DB FF     .      STA PARM1      ;SAVE CONTROL FLAG
6609     17F5     3A C1 FF     .      LDA CURCOL    ;GET THE CURRENT COLUMN
6610     17F8     3D . .     .      DCR A         ;SET TO FIND PREVIOUS COLUMN
6611     17F9     . . .      ;*****
6612     17F9     . . .      ; ROM BREAK 3
6613     17F9     C3 02 18     .      JMP ZBRK3C
6614     17FC     . . .      ORG ZBRK2+40000
6615     1800     . . .      ZBRK3 EQU $
6616     1800     . . .      ; THIS ROM WAS MODIFIED FOR MULTIPOINT
6617     1800     . . .      ; COMPATIBILITY. ONLY ROUTINE CHANGED WAS
6618     1800     . . .      ; 'STTERM'
6619     1800     55 . .     .      DB VERSN2     ;ROM VERSION FLAG
6620     1801     18 . .     .      DB ZBRK3/256
6621     1802     . . .      ZBRK3C EQU $
6622     1802     . . .      ;*****
6623     1802     CD 10 08     .      CALL RCADRO   ;PREVIOUS COLUMN PRESENT?
6624     1805     FA 14 48     .      JM ZBELL      ;NO - SOUND BELL AND RETURN
6625     1808     3A C5 FF     .      LDA LSTFMT   ;YES - RECALL LAST FORMAT CT
6626     180B     FE C0 .     .      CPI STPR      ;WAS IT A START PROTECT?
6627     180D     C4 E6 17     .      CNZ PRSTRT   ;NO - ENTER STPR
6628     1810     3A DB FF     .      LDA PARM1    ;RECALL FORMAT CONTROL FLAG
6629     1813     47 . .     .      MOV B,A      ;TO BE STORED
6630     1814     . . .      ;
6631     1814     . . .      ; ENTER THE FORMAT CONTROL FLAG
6632     1814     . . .      ;
6633     1814     . . .      PRO100 EQU $
6634     1814     CD CF 1A     .      CALL CHKFMS   ;FORMAT MODE?
6635     1817     C0 . .     .      RNZ          ;YES - TERMINATE
6636     1818     78 . .     .      MOV A,B      ;NO - ADD CHAR TO DISPLAY
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 199
=====
6637     1819     F5 . .      PUSH PSW          ;SAVE THE CONTROL CODE
6638     181A     CD 43 24    CALL DISPC1      ;(DISPC1 DESTROYS "LSTFMT"
6639     181D     F1 . .      POP PSW          ;RECALL CONTROL CODE
6640     181E     32 C5 FF    STA LSTFMT      ;NEW ENTRY
6641     1821     C9 . .      RET
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 200
=====
6643     1822     . . .      ;
6644     1822     . . .      ; ENTREN - ENABLE ENTER VIA ESCAPE SEQUENCE
6645     1822     . . .      ;
6646     1822     . . .      ENTREN EQU $
6647     1822     01 00 40    LXI B,SENER ;SET DISPLAY SEND PENDING
6648     1825     . . .      ; FALL INTO "SBLXF0"
6649     1825     . . .      ;*****
6650     1825     . . .      ; SBLXFU - SET BLOCK TRANSFER FLAG FOR ESCAPE *
6651     1825     . . .      ; SEQUENCE INITIATED BLOCK TRANSFERS *
6652     1825     . . .      ;*****
6653     1825     . . .      ;
6654     1825     . . .      ; ENTRY: B = FLAG TO BE SET IN "MFLGS"
6655     1825     . . .      ; C = FLAG TO BE SET IN "MFLGS2"
6656     1825     . . .      ;
6657     1825     . . .      ; EXIT : ALL REGISTERS DESTROYED
6658     1825     . . .      ; X-ON AND DC2 PENDING FLAGS ARE SET
6659     1825     . . .      ; ACCORDING TO THE SETTINGS OF G AND H
6660     1825     . . .      ;
6661     1825     . . .      SBLXF0 EQU $
6662     1825     CD B9 11    CALL CLRXON ;CLEAR BLOCK TRANSFER TRIGGE
6663     1828     . . .      ;
6664     1828     . . .      ; SBLXFA - DETERMINE DC2 HANDSHAKE MODE FOR
6665     1828     . . .      ; NON-BLOCK MODE KEYBOARD INITIATED BLOCK
6666     1828     . . .      ; TRANSFERS
6667     1828     . . .      ;
6668     1828     . . .      SBLXFA EQU $
6669     1828     3A FB FF    LDA KBJMPR ;GET THE STRAP SETTINGS
6670     1828     E6 40 .    ANI HNDSHK ;DC2 ON ALL BLOCK TRANSFERS?
6671     182D     CA 3D 18    JZ SBL010 ;NO - DO NOT SET DC2 FLAG
6672     1830     . . .      ; YES - FALL INTO "SBLXF1"
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 201
6674	1830	.	.	*****	
6675	1830	.	.	; SBLXF1 - SET BLOCK TRANSFER FLAG FOR KEYBOARD *	
6676	1830	.	.	; INITIATED BLOCK TRANSFERS *	
6677	1830	.	.	*****	
6678	1830	.	.	;	
6679	1830	.	.	; ENTRY: B = FLAG TO BE SET IN "MFLGS"	
6680	1830	.	.	; C = FLAG TO BE SET IN "MFLGS2"	
6681	1830	.	.	;	
6682	1830	.	.	SBLXF1 EQU \$	
6683	1830	3A	FB FF	LDA KBJMPR ;GET THE STRAP SETTINGS	
6684	1833	E6	80 .	ANI DC2SND ;INHIBIT DC2 HANDSHAKE?	
6685	1835	3E	01 .	MVI A,SDC2/256 ;(SET DC2 PENDING FLAG)	
6686	1837	CA	3E 18	JZ SBL020 ;NO - SET DC2 PENDING FLAG	
6687	183A	CD	CD 04	CALL CHKCT1 ;YES - SET BLOCK TRANSFER	
6688	183D	.	.	; TRIGGER TO CAUSE IMMEDIATE	
6689	183D	.	.	; TRANSMISSION OF DATA	
6690	183D	.	.	SBL010 EQU \$	
6691	183D	78	.	MOV A,B ;PUT FLAG INTO A-REGISTER	
6692	183E	.	.	SBL020 EQU \$	
6693	183E	B0	.	ORA B ;ADD IN OPTIONAL DC2 FLAG	
6694	183F	47	.	MOV B,A ;SAVE FLAGS IN B-REGISTER	
6695	1840	CD	95 11	CALL CKRMTE ;REMOTE MODE ENABLED?	
6696	1843	C8	.	RZ ;NO - DON'T SET BLOCK XFR	
6697	1844	21	70 FF	LXI H,MFLGS ;YES - SET DATA PENDING	
6698	1847	78	.	MOV A,B ;FLAGS	
6699	1848	B6	.	ORA M	
6700	1849	77	.	MOV M,A	
6701	184A	2B	.	DCX H	
6702	184B	79	.	MOV A,C	
6703	184C	B6	.	ORA M ;SET FLAG IN "MFLGS2"	
6704	184D	77	.	MOV M,A	
6705	184E	C3	5E 17	JMP KBLOK ;DISABLE THE KEYBOARD	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 202
=====
6707     1851      . . .      ;*****
6708     1851      . . .      ; SDTRM1 - SEND TERMINATOR CHARACTER *
6709     1851      . . .      ;*****
6710     1851      . . .      ;
6711     1851      . . .      ; EXIT : A DESTROYED
6712     1851      . . .      ;
6713     1851      . . .      SDTRM1 EQU $
6714     1851      CD 84 11      CALL GTMODE      ;PAGE MODE?
6715     1854      3A 04 50      LDA BLKTRM      ;(GET BLOCK TERMINATOR)
6716     1857      C2 22 19      JNZ XPUTDC      ;YES - SEND BLOCK TERM ONLY
6717     185A      . . .      SDTRM2 EQU $      ;NO - SEND CR(LF)
6718     185A      3E 0D .      MVI A,CR
6719     185C      CD 22 19      CALL XPIJDC      ;TRANSMIT RETURN
6720     185F      3A F3 FF      LDA MDFLG2
6721     1862      E6 04 .      ANI AUTOLF      ;AUTO LINE FEED ENABLED?
6722     1864      C8 . .      RZ              ;NO - RETURN
6723     1865      . . .      SDTRM3 EQU $
6724     1865      3E 0A .      MVI A,LF      ;YES - TRANSMIT LINE FEED
6725     1867      C3 22 19      JMP XPUTDC
=====

```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 203
=====
6727     186A      . . .      ;*****
6728     186A      . . .      ; SETDFL - SET DATA TRANSFER FLAG *
6729     186A      . . .      ;*****
6730     186A      . . .      ;
6731     186A      . . .      ; ENTRY:  A = FLAG BIT TO BE SET
6732     186A      . . .      ;
6733     186A      . . .      ; EXIT :  H = BASEH
6734     186A      . . .      ;          A,L DESTROYED
6735     186A      . . .      ;
6736     186A      . . .      SETDFU EQU $          ;SET DATA COMM INPUT FLAG
6737     186A      3E 01 .      MVI  A,SDACUM        ;SET FLAG BIT TO BE SET
6738     186C      . . .      SETDFL EQU $
6739     186C      21 6E FF      LXI  H,DFLGS
6740     186F      86 . .      ORA  M              ;MERGE FLAG BIT TO EXISTING
6741     1870      77 . .      MOV  M,A            ;FLAGS
6742     1871      C9 . .      RET                ;RETURN
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 204
=====
6744     1872      . . .      ;
6745     1872      . . .      ; * * * * *
6746     1872      . . .      ;
6747     1872      . . .      ; SETLFT,SETRHT - SET LEFT AND RIGHT MARGINS
6748     1872      . . .      ;
6749     1872      . . .      ; ENTRY:  H = BASEH
6750     1872      . . .      ; CURCOL = CURSOR COLUMN POSITION
6751     1872      . . .      ;
6752     1872      . . .      ; EXIT :  LFTMGN,RHTMGN SET APPROPRIATELY
6753     1872      . . .      ;
6754     1872      . . .      ; SETLFT EQU $
6755     1872      CD D4 1A    CALL CHKFMT      ;FORMAT MODE?
6756     1875      C0 . .     RNZ              ;YES - DON'T SET MARGIN
6757     1876      3A BE FF    LDA RHTMGN      ;NO - GET THE RIGHT MARGIN
6758     1879      2E C1 .     MVI L,CURCOL-BASE
6759     187B      BE . .     CMP M           ;CURSOR AFTER RIGHT MARGIN?
6760     187C      FA C3 25   JM DSPCH1       ;YES - DON'T SET MARGIN
6761     187F      7E . .     MOV A,M         ;NO - SET NEW LEFT MARGIN
6762     1880      32 BF FF    STA LFTMGN
6763     1883      C9 . .     RET            ;RETURN
6764     1884      . . .      ;
6765     1884      . . .      ; SETRHT EQU $
6766     1884      CD D4 1A    CALL CHKFMT      ;FORMAT MODE?
6767     1887      C0 . .     RNZ              ;YES - DON' SET MARGIN
6768     1888      3A C1 FF    LDA CURCOL      ;GET CURRENT CURSOR COLUMN
6769     188B      2E BF .     MVI L,LFTMGN-BASE
6770     188D      BE . .     CMP M           ;BEFORE LEFT MARGIN?
6771     188E      FA C3 25   JM DSPCH1       ;YES - DON'T SET MARGIN
6772     1891      2B . .     DCX H          ;NO - SET NEW RIGHT MARGIN
6773     1892      77 . .     MOV M,A
6774     1893      C9 . .     RET            ;RETURN
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 205
=====
6776     1894      . . .      ;*****
6777     1894      . . .      ; SETMF2 - SET FLAG BIT IN MFLGS2 *
6778     1894      . . .      ;*****
6779     1894      . . .      ;
6780     1894      . . .      ; ENTRY:  A = FLAG BIT TO BE SET
6781     1894      . . .      ;
6782     1894      . . .      ; EXIT :  A = UPDATED MFLGS2 VALUE
6783     1894      . . .      ;           H,L = MFLGS2
6784     1894      . . .      ;
6785     1894      . . .      SETMF2 EQU  $
6786     1894      21 6F FF      LXI  H,MFLGS2
6787     1897      86 . .      ORA  M           ;ADD BIT TO MFLGS2
6788     1898      77 . .      MOV  M,A        ;STORE NEW SETTINGS
6789     1899      C9 . .      RET           ;RETURN
=====
```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 206
=====
6791     189A      . . .      ;*****
6792     189A      . . .      ; SETTRM - SET NON-DISPLAYING TERMINATOR *
6793     189A      . . .      ;*****
6794     189A      . . .      SETTRM EQU $
6795     189A      3E 01 .      MVI A,IGNTRM ;SET TO IGNORE NON-DISPLAYIN
6796     189C      32 6D FF      STA TRMFCT   ;TERMINATORS
6797     189F      3E C4 .      MVI A,STPFLG ;ADD NON-DISPLAYING
6798     18A1      CD 45 24      CALL DISPC2  ;TERMINATOR TO DISPLAY
6799     18A4      C3 96 1F      JMP FLDSRX   ;SET "LSTCOL" TO MAXCOL+1
6800     18A7      . . .      ;           TO FORCE LINE RE-SCAN TO
6801     18A7      . . .      ;           INHIBIT DELETION OF NEW
6802     18A7      . . .      ;           NON-DISPLAYING TERMINATOR
=====
```

					PAGE 207	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS		
6804	18A7	.	.	.	;*****	
6805	18A7	.	.	.	; SNDCDE - SEND ATTENTION/FUNCTION CODE *	
6806	18A7	.	.	.	;*****	
6807	18A7	.	.	.	SNDCDE EQU \$	
6808	18A7	3A	6E	FF	LDA	DFLGS ;GET DATA TRANSFER FLAGS
6809	18AA	E6	01	.	ANI	SDACOM ;COMMAND FROM DATA COMM?
6810	18AC	C0	.	.	RNZ	;YES - IGNORE IT
6811	18AD	21	67	7F	LXI	H,SNDCDB ;SET TO ACCUMULATE OCTAL
6812	18B0	3E	08	.	MVI	A,OCTRDY ;CODE CHARACTER
6813	18B2	C3	34	05	JMP	ESCAP0
6814	18B5	.	.	.	;*****	
6815	18B5	.	.	.	; <A> - SEND ATTENTION CODE *	
6816	18B5	.	.	.	;*****	
6817	18B5	.	.	.	SNDCD1 EQU \$	
6818	18B5	3A	DE	FF	LDA	IODATA ;GET ACCUMULATED VALUE
6819	18B8	47	.	.	MOV	B,A ;PUT CODE INTO B-REGISTER
6820	18B9	3E	0B	.	MVI	A,SNDBTN ;SET DATA COMM CONTROL CODE
6821	18BB	C3	73	13	JMP	DCMCTL ;PERFORM FUNCTION

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 208
6823	18BE	.	.	;	*****
6824	18BE	.	.	; STRTBL - SET FIRST DISPLAY OUT CHARACTER FOR *	
6825	18BE	.	.	; BLOCK STORE	*
6826	18BE	.	.	;	*****
6827	18BE	.	.	; CALLED ONLY BY IO CODE BEFORE 'RECORD' OPERATION	
6828	18BE	.	.	;	*****
6829	18BE	.	.	;	
6830	18BE	.	.	; ENTRY: DON'T CARE	
6831	18BE	.	.	;	
6832	18BE	.	.	; EXIT : CURCOL,CURROW = STARTING POSITION	
6833	18BE	.	.	;	
6834	18BE	.	.	; IF THE AUTO TERMINATOR STRAP (J) IS OUT, A	
6835	18BE	.	.	; TERMINATOR IS PLACED AHEAD OF THE CURRENT	
6836	18BE	.	.	; CURSOR POSITION AND A REVERSE SCAN IS MADE	
6837	18BE	.	.	; FOR THE FIRST TERMINATOR BEFORE THE CURRENT	
6838	18BE	.	.	; CURSOR POSITION. OTHERWISE, THE CURSOR IS	
6839	18BE	.	.	; PLACED AT THE HOME POSITION	
6840	18BE	.	.	;	
6841	18BE	.	.	STRTBL EQU \$	
6842	18BE	CD	C4 18	CALL STRTB1 ;SET CURSOR START POSITION	
6843	18C1	C3	2E 7D	JMP INITDG ;SET UP DISPLAY GET ROUTINE	
6844	18C4	.	.	;	
6845	18C4	.	.	STRTB1 EQU \$	
6846	18C4	.	.	;	*****
6847	18C4	CD	23 60	CALL ZMUCHK ;AUTO PLOT MENU UP?	
6848	18C7	C2	6A 60	JNZ ZAPHME ;YES, HOME AUTO PLOT CURSOR	
6849	18CA	.	.	;	*****
6850	18CA	3A	FA FF	LDA KBJMP2 ;GET KEYBOARD JUMPERS 2	
6851	18CD	E6	01 .	ANI AUTTRM ;AUTO TERMINATOR ENABLED?	
6852	18CF	CA	55 19	JZ XMOHME ;NO - HOME THE CURSOR	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 209
6854	18D2	.	.	*****	
6855	18D2	.	.	; THIS ROUTINE WAS MODIFIED FOR MULTIPOINT	
6856	18D2	.	.	; COMPATIBILITY ON 17AUG77	
6857	18D2	.	.	*****	
6858	18D2	.	.	*****	
6859	18D2	.	.	; STTERM - SET AUTO TERMINATOR *	
6860	18D2	.	.	*****	
6861	18D2	.	.	;	
6862	18D2	.	.	; EXIT : Z => AUTO TERMINATOR NOT SET	
6863	18D2	.	.	; NZ => AUTO TERMINATOR SET	
6864	18D2	.	.	;	
6865	18D2	.	.	STTERM EQU \$	
6866	18D2	3E	FB	MVI A,377Q-NOSEND	
6867	18D4	CD	53	CALL CLRDFL ;CLEAR NO DATA FLAG	
6868	18D7	F6	08	ORI SKPTRM ;SET TO SKIP INITIAL BLOCK	
6869	18D9	77	.	MOV M,A ;TERMINATOR CHARACTER	
6870	18DA	CD	D4	CALL CHKFMT ;FORMAT MODE ENABLED?	
6871	18DD	EE	08	XRI FORMAT ;REVERSE FORMAT SENSE	
6872	18DF	C4	DA	CNZ CHKMLK ;CONDITIONAL MEM LOCK ENABLE	
6873	18E2	21	AC	LXI H,FRBLKS ;GET LSB OF FREE BLKS PTR	
6874	18E5	B6	.	ORA M ;ANY FREE BLOCKS?	
6875	18E6	CA	10	JZ MLOCK ;NO - FORCE MEMORY LOCK ON	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 210
=====
6877     18E9      . . .      ;*****
6878     18E9      . . .      ; SPACE AVAILABLE - STORE NON-DISPLAYING *
6879     18E9      . . .      ;   TERMINATOR AT CURRENT CURSOR POSITION *
6880     18E9      . . .      ;*****
6881     18E9      . . .      STB050 EQU $
6882     18E9      CD 9A 18      CALL SETTRM      ;STORE TERMINATOR
6883     18EC      26 C4 .      MVI H,STPFLG    ;SET SEARCH PREV TERMINATOR
6884     18EE      BC . . .      CMP H            ;ALREADY AT TERMINATOR?
6885     18EF      CA 15 19      JZ STB090       ;YES NULL RECORD SEND
6886     18F2      3A 04 50      LDA BLKTRM      ;GET BLOCK TERMINATOR CHAR
6887     18F5      6F . . .      MOV L,A         ;SET PARAMETERS FOR REVERSE
6888     18F6      CD 5E 19      CALL BACKT1     ;IS THER A PREV TERMINATOR?
6889     18F9      C2 04 19      JNZ STB080      ;NO - HOME THE CURSOR
6890     18FC      CD A1 07      CALL RCADRA     ;DOES THE CHARACTER EXIST?
6891     18FF      C4 22 22      CNZ CRLF        ;NO - START AT NEXT LINE
6892     1902      B4 . . .      ORA H           ;SET NZ RET
6893     1903      C9 . . .      RET
6894     1904      . . .      ;*****
6895     1904      . . .      ; NO PREVIOUS TERMINATOR - HOME THE CURSOR *
6896     1904      . . .      ;*****
6897     1904      . . .      STB080 EQU $
6898     1904      2A C0 FF      LHLD CURROW     ;SAVE THE CURRENT ROW AND
6899     1907      E5 . . .      PUSH H          ;COLUMN VALUES
6900     1908      CD 66 14      CALL DPSEN1     ;HOME CURSOR FOR TRANSMISSIO
6901     190B      2A C0 FF      LHLD CURROW     ;GET NEW ROW AND COLUMN
6902     190E      C1 . . .      POP B           ;RECALL OLD ROW AND COLUMN
6903     190F      7C . . .      MOV A,H         ;COMPARE TO HOME ROW AND
6904     1910      90 . . .      SUB B           ;COLUMN
6905     1911      C0 . . .      RNZ             ;RET NO EQ
6906     1912      7D . . .      MOV A,L
6907     1913      91 . . .      SUB C
6908     1914      C0 . . .      RNZ             ;RET NOT EQ
6909     1915      . . .      STB090 EQU $
6910     1915      3E 04 .      MVI A,NOSEND   ;NO - SET FOR NO DATA
6911     1917      C3 6C 18      JMP SETDFL     ;RETURN
6912     191A      00 00 .      DB 0,0         ;PATCH FREE SPACE
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 211
6914	191C	.	.	*****	
6915	191C	.	.	; XPUTDC - TRANSMIT CHARACTER *	
6916	191C	.	.	*****	
6917	191C	.	.	;	
6918	191C	.	.	; ENTRY: A = CHARACTER TO BE TRANSMITTED	
6919	191C	.	.	;	
6920	191C	.	.	; EXIT : NC - TRANSMIT SUCCESSFUL	
6921	191C	.	.	; C - TRANSMIT FAILED	
6922	191C	.	.	; A DESTROYED	
6923	191C	.	.	;	
6924	191C	.	.	ESCOU EQU \$;OUTPUT AN ESCAPE CODE	
6925	191C	3E	1B	MVI A,ESC	
6926	191E	CD	22 19	CALL XPUTDC	
6927	1921	78	.	MOV A,B ;FOLLOWED BY CHAR IN B-REG	
6928	1922	.	.	;	
6929	1922	.	.	XPUTDC EQU \$	
6930	1922	B7	.	ORA A ;SET C-FLAG FALSE	
6931	1923	F5	.	PUSH PSW ;SAVE THE FLAGS AND A-REG	
6932	1924	CD	95 11	CALL CKRMTE ;REMOTE MODE ENABLED?	
6933	1927	CA	35 19	JZ XPD005 ;NO - EXIT	
6934	192A	.	.	XPD001 EQU \$	
6935	192A	F1	.	POP PSW ;YES - RECALL THE CHARACTER	
6936	192B	F5	.	PUSH PSW ;SAVE CONTENTS OF A AND FLAG	
6937	192C	CD	1A 50	CALL ZPUTDC ;TRANSMIT THE CHAR IN A-REG	
6938	192F	DA	4B 19	JC XPD050 ;ERROR - REPORT IT	
6939	1932	C2	37 19	JNZ XPD010 ;WAIT - TRY AGAIN	
6940	1935	.	.	XPD005 EQU \$	
6941	1935	F1	.	POP PSW ;DONE - RECALL FLAGS AND CHA	
6942	1936	C9	.	RET ;RETURN	
6943	1937	.	.	;	
6944	1937	.	.	;	
6945	1937	.	.	*****	
6946	1937	.	.	; WAIT FOR DATACOM - RETRY OPERATION *	
6947	1937	.	.	*****	
6948	1937	.	.	XPD010 EQU \$	
6949	1937	E5	.	PUSH H ;SAVE THE REGISTERS	
6950	1938	D5	.	PUSH D	
6951	1939	C5	.	PUSH B	
6952	193A	CD	D8 16	CALL IOCTMN ;MONITOR CARTRIDGE TAPES	
6953	193D	3E	0A .	MVI A,CKBRKY ;LOOK FOR A BREAK KEY HIT	
6954	193F	CD	08 48	CALL ZKBCTL ;BREAK KEY HIT?	
6955	1942	C1	.	POP B ;(RESTORE REGISTERS)	
6956	1943	D1	.	POP D	
6957	1944	E1	.	POP H	
6958	1945	CA	2A 19	JZ XPD001 ;NO - TRY TO OUTPUT AGAIN	
6959	1948	CD	6C 13	CALL BRKDC ;YES - BREAK DATA COMM	
6960	194B	.	.	;	
				FALL INTO ERROR EXIT ROUTINE	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 212
=====
6962     194B      . . .      ;*****
6963     194B      . . .      ; DATA COMM ERROR DETECTED - REPORT ERROR *
6964     194B      . . .      ;*****
6965     194B      . . .      XPD050 EQU $
6966     194B      33 . .      INX SP      ;RESTORE STACK LEVEL WITHOUT
6967     194C      33 . .      INX SP      ;AFFECTING THE FLAGS
6968     194D      C2 85 13     JNZ HANGUO   ;FATAL - HANG THE TERMINAL
6969     1950      CD 14 48     CALL ZBELL   ;NON-FATAL - SOUND BELL
6970     1953      37 . .      STC          ;RETURN FAIL (C-FLAG = TRUE)
6971     1954      C9 . .      RET
```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 213
=====
6973     1955      . . .      ;*****
6974     1955      . . .      ; XMOHME - HOME CURSOR INCLUDING TRANSMIT *
6975     1955      . . .      ;   ONLY FIELDS                               *
6976     1955      . . .      ;*****
6977     1955      . . .      XMOHME EQU $
6978     1955      CD 6A 18    CALL SETDF0      ;SET DATA COMM INPUT FLAG
6979     1958      . . .      ;                               TO ENABLE TRANSMIT ONLY
6980     1958      . . .      ;                               FIELDS
6981     1958      C3 A2 1E    JMP  CURPH1      ;HOME THE CURSOR
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 214
6983	195B	.	.	. ;*****	
6984	195B	.	.	. ; BACKT1 - LOCATE PREVIOUS CHARACTER *	
6985	195B	.	.	. ;*****	
6986	195B	.	.	. ;	
6987	195B	.	.	. ; ENTRY: IODATA = CHARS TO FIND (2 BYTES)	
6988	195B	.	.	. ; CURCOL,CURROW = CURRENT CURSOR POSITION	
6989	195B	.	.	. ;	
6990	195B	.	.	. ; EXIT : Z - CHARACTER FOUND	
6991	195B	.	.	. ; DISPLAY AND CURSOR SET TO CHARACTER	
6992	195B	.	.	. ; POSITION IN DISPLAY MEMORY - ALL	
6993	195B	.	.	. ; DISPLAY PARAMETERS UPDATED	
6994	195B	.	.	. ; NZ - CHARACTER NOT FOUND	
6995	195B	.	.	. ; DISPLAY UNCHANGED	
6996	195B	.	.	. ; ALL REGISTERS DESTROYED	
6997	195B	.	.	. ;	
6998	195B	.	.	. BACKT0 EQU \$;LOOK FOR PREVIOUS FIELD	
6999	195B	21	C1	C1 LXI H,ENDPR*256+ENDPR	
7000	195E	.	.	. BACKT1 EQU \$	
7001	195E	22	D7	FF SHLD LCHKSM ;SAVE CHARACTERS TO BE FOUND	
7002	1961	AF	.	. XRA A ;CLEAR ROLL COUNT	
7003	1962	32	82	FF STA ROLLCT	
7004	1965	2A	C0	FF LHLD CURROW ;SAVE THE CURRENT STATE OF	
7005	1968	22	DE	FF SHLD LDATA ;THE DISPLAY	
7006	1968	2A	C9	FF LHLD LSTLIN	
7007	196E	22	D5	FF SHLD LADDR	
7008	1971	3E	01	. MVI A,IGNTRM ;SET TO IGNORE NON-DISPLAYIN	
7009	1973	32	6D	FF STA TRMFACT ;TERMINATOR	
7010	1976	CD	B1	07 CALL RCADR1 ;DOES THE CURRENT LINE EXITS	
7011	1979	3A	DF	FF LDA LDATA+1 ;(RECALL CURRENT COLUMN)	
7012	197C	F2	94	19 JP BKT230 ;YES - SEARCH FOR PREV FIELD	
7013	197F	3A	C0	FF LDA CURROW ;NO - LOCATE LAST LINE	
7014	1982	21	6B	FF LXI H,MLKROW ;CURRENT ROW LESS THAN	
7015	1985	BE	.	. CMP M ;MEMORY LOCK ROW?	
7016	1986	F2	D3	19 JP BKT300 ;NO - LOOK FOR UNLOCKED LINE	
7017	1989	.	.	. BKT210 EQU \$;YES - START FROM LAST LINE	
7018	1989	CD	38	12 CALL CURPHD	
7019	198C	3A	C7	FF LDA LSTROW ;FORCE TO LAST ALLOCATED	
7020	198F	32	C0	FF STA CURROW ;ROW	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 215
7022	1992	.	.	.	;*****	
7023	1992	.	.	.	; LOCATE THE LAST FIELD IN THE LINE *	
7024	1992	.	.	.	;*****	
7025	1992	.	.	.	BKT220 EQU \$	
7026	1992	3E	4F	.	MVI A,MAXCOL ;SET SEARCH LIMIT	
7027	1994	.	.	.	BKT230 EQU \$	
7028	1994	32	85	FF	STA TMPCOL ;SAVE THE SEARCH LIMIT	
7029	1997	2A	C9	FF	LHLD LSTLIN ;GET SEARCH START ADDRESS	
7030	199A	E8	.	.	XCHG ;PUT ADDRESS IN D,E	
7031	199B	2A	D7	FF	LHLD LCHKSM ;RECALL CHARS TO BE FOUND	
7032	199E	CD	DA	20	CALL FNDLST ;ANY FIELDS IN LINE?	
7033	19A1	F2	E9	19	JP BKT400 ;YES - SET DISPLAY TO FIELD	
7034	19A4	3A	6B	FF	LDA MLKROW ;NO - SEE IF TOP UNLOCKED	
7035	19A7	21	C0	FF	LXI H,CURROW ;LINE HAS BEEN REACHED	
7036	19AA	BE	.	.	CMP M ;REACHED MEMORY LOCK ROW?	
7037	19AB	CA	DF	19	JZ BKT310 ;YES - CONTINUE ABOVE DISPLA	
7038	19AE	3A	82	FF	LDA ROLLCT	
7039	19B1	B6	.	.	ORA M ;ROLL COUNT AND ROW = ZERO?	
7040	19B2	CA	49	1A	JZ BKT500 ;YES - NO PREVIOUS FIELD IN	
7041	19B5	.	.	.	; LOCKED AREA, RESTORE DISPL	
7042	19B5	35	.	.	DCR M ;NO - MOVE TO PREVIOUS ROW	
7043	19B6	2E	82	.	MVI L,ROLLCT	
7044	19B8	7E	.	.	MOV A,M ;GET ROLL COUNT	
7045	19B9	B7	.	.	ORA A ;SEARCHING ABOVE DISPLAY?	
7046	19BA	CA	C1	19	JZ BKT240 ;NO - DUN'T INCREMENT COUNT	
7047	19BD	34	.	.	INR M ;ROLL OVERFLOW?	
7048	19BE	CA	49	1A	JZ BKT500 ;YES - RESTORE DISPLAY	
7049	19C1	.	.	.	BKT240 EQU \$;NO - LOOK TO PREVIOUS LINE	
7050	19C1	2A	C9	FF	LHLD LSTLIN ;RECALL CURRENT LINE ADDR	
7051	19C4	.	.	.	BKT250 EQU \$	
7052	19C4	23	.	.	INX H ;GET ADDRESS OF PREVIOUS	
7053	19C5	23	.	.	INX H ;LINE	
7054	19C6	CD	C6	1A	CALL CHAIN ;GET PREVIOUS LINE ADDRESS	
7055	19C9	B7	.	.	ORA A ;DOES PREVIOUS LINE EXIST?	
7056	19CA	CA	49	1A	JZ BKT500 ;NO - RESTORE DISPLAY	
7057	19CD	22	C9	FF	SHLD LSTLIN ;YES - SAVE ADDRESS OF LINE	
7058	19D0	C3	92	19	JMP BKT220 ;LOCATE LAST FIELD IN LINE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 216
=====
7060     19D3     . . .      ;
7061     19D3     . . .      ; ROW NOT FOUND AND CURSOR BELOW MEMORY LOCK
7062     19D3     . . .      ; LINE - LOCATE LAST LINE TO START
7063     19D3     . . .      ;
7064     19D3     . . .      BKT300 EQU $
7065     19D3     93 . .     SUB E          ;(E = # OF ROWS TO LAST LN
7066     19D4     BE . .     CMP M          ;LAST ROW BELOW LOCKED AREA?
7067     19D5     F2 89 19   JP BKT210      ;YES - START AT LAST LINE
7068     19D8     7E . .     MOV A,M        ;NO - SEARCH ABOVE DISPLAY
7069     19D9     32 C0 FF   STA CURROW     ;SET "CURROW" TO MEM LOCK RO
7070     19DC     21 C0 FF   LXI H,CURROW  ;SET H,L -> "CURROW"
7071     19DF     . . .      ;
7072     19DF     . . .      ; NO PREVIOUS FIELDS ON DISPLAY - LOOK ABOVE DISP
7073     19DF     . . .      ;
7074     19DF     . . .      BKT310 EQU $
7075     19DF     35 . .     DCR M          ;DECREMENT ROW NUMBER
7076     19E0     2E 82 .    MVI L,ROLLCT-BASE
7077     19E2     34 . .     INR M          ;INCREMENT ROLL COUNT
7078     19E3     2A CB FF   LHLD TOPLIN   ;GET TOP DISPLAY LINE ADDR
7079     19E6     C3 C4 19   JMP BKT250    ;LOOK FOR PREVIOUS ROW
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 217
7081	19E9	.	.	.	;
7082	19E9	.	.	.	; FIELD FOUND - SET DISPLAY
7083	19E9	.	.	.	;
7084	19E9	.	.	.	BKT400 EQU \$
7085	19E9	2A	D5	FF	LHLD LADDR ;RESTORE VALUE OF LSTLIN
7086	19EC	EB	.	.	XCHG ;AND SAVE ADDRESS OF
7087	19ED	2A	C9	FF	LHLD LSTLIN ;LINE WHERE FIELD IS
7088	19F0	22	D5	FF	SHLD LADDR
7089	19F3	EB	.	.	XCHG
7090	19F4	22	C9	FF	SHLD LSTLIN
7091	19F7	3A	85	FF	LDA TMPCOL ;COMPUTE COLUMN LOCATION
7092	19FA	90	.	.	SUB B
7093	19FB	32	C1	FF	STA CURCOL
7094	19FE	3A	82	FF	LDA ROLLCT
7095	1A01	B7	.	.	ORA A ;ROW ABOVE DISPLAY?
7096	1A02	CA	34	1A	JZ BKT450 ;NO - EXIT
7097	1A05	.	.	.	;
7098	1A05	.	.	.	; ROW ABOVE DISPLAY ROLL IT DOWN
7099	1A05	.	.	.	;
7100	1A05	.	.	.	BKT410 EQU \$
7101	1A05	3E	E8	.	MVI A,-MAXROW-1 ;COMPUTE NUMBER OF LINES
7102	1A07	21	6B	FF	LXI H,MLKROW ;TO ROLL FOR ONE PAGE
7103	1A0A	86	.	.	ADD M
7104	1A0B	32	82	FF	STA ROLLCT ;SAVE ROLL COUNT
7105	1A0E	.	.	.	BKT420 EQU \$
7106	1A0E	CD	CE	0C	CALL ROLLDN ;ROLL DOWN ONE LINE
7107	1A11	CA	2A	1A	JZ BKT430 ;ROLL FAIL - CHECK FOR FIELD
7108	1A14	21	C0	FF	LXI H,CURROW
7109	1A17	34	.	.	INR M ;INCREMENT ROW NUMBER
7110	1A18	2E	82	.	MVI L,ROLLCT
7111	1A1A	34	.	.	INR M ;PAGE COMPLETED?
7112	1A1B	C2	0E	1A	JNZ BKT420 ;NO - CONTINUE ROLLING
7113	1A1E	3A	C0	FF	LDA CURROW
7114	1A21	2E	6B	.	MVI L,MLKROW
7115	1A23	96	.	.	SUB M ;IS DESIRED ROW ON SCREEN?
7116	1A24	FA	05	1A	JM BKT410 ;NO - ROLL DOWN ANOTHER PAGE
7117	1A27	C3	34	1A	JMP BKT450 ;YES - EXIT
7118	1A2A	.	.	.	;
7119	1A2A	.	.	.	; ROLL FAILED - CHECK FOR FIELD ON SCREEN
7120	1A2A	.	.	.	;
7121	1A2A	.	.	.	BKT430 EQU \$
7122	1A2A	3A	C0	FF	LDA CURROW ;GET CURRENT ROW NUMBER
7123	1A2D	21	6B	FF	LXI H,MLKROW ;SUBTRACT MEMRY LOCK RWS
7124	1A30	96	.	.	SUB M ;IS FIELD ON SCREEN?
7125	1A31	FA	4F	1A	JM BKT510 ;NO - RESTORE DISPLAY, ROLL
7126	1A34	.	.	.	DOWN FAILED BECAUSE OF NO
7127	1A34	.	.	.	MEMORY TO FILL TO MEMORY
7128	1A34	.	.	.	LOCK LINE

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 218
=====
7130     1A34      . . .      ;
7131     1A34      . . .      ; FIELD ON SCREEN - SET SCREEN VALUES
7132     1A34      . . .      ;
7133     1A34      . . .      BKT450 EQU $
7134     1A34      3A C0 FF      LDA  CURROW ;SET LAST ROW VALUE TO
7135     1A37      32 C7 FF      STA  LSTROW ;ROW FOUND
7136     1A3A      AF . .      XRA  A      ;SET LAST COL DONE TO ZERO
7137     1A3B      32 C8 FF      STA  LSTCOL
7138     1A3E      2A D5 FF      LHLD LADDR  ;SET ADDRESSES TO LOCATION
7139     1A41      . . .      BACKT5 EQU $
7140     1A41      22 C9 FF      SHLD LSTLIN ;OF FIELD
7141     1A44      2B . .      DCX  H      ;SET CURADR TO CORRESPOND
7142     1A45      22 C3 FF      SHLD CURADR
7143     1A48      C9 . .      RET        ;RETURN
7144     1A49      . . .      ;
7145     1A49      . . .      ; FIELD NOT FOUND - RESTORE DISPLAY
7146     1A49      . . .      ;
7147     1A49      . . .      BKT500 EQU $
7148     1A49      2A D5 FF      LHLD LADDR ;RESTORE LAST LINE ADDRESS
7149     1A4C      22 C9 FF      SHLD LSTLIN
7150     1A4F      . . .      BKT510 EQU $
7151     1A4F      2A DE FF      LHLD LDATA ;RESTORE CURRENT ROW AND
7152     1A52      22 C0 FF      SHLD CURROW ;COLUMN
7153     1A55      F6 FF .      ORI  377Q  ;SET Z FALSE
7154     1A57      C9 . .      RET        ;RETURN NOT FOUND (NZ)
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 219
7156	1A58	.	.	. ;*****	
7157	1A58	.	.	. ; BKTAB - BACK TAB *	
7158	1A58	.	.	. ;*****	
7159	1A58	.	.	. BKTAB EQU \$	
7160	1A58	CD	CB	1A CALL CHKFM0 ;FORMAT/SOFT KEY DEFINE MODE	
7161	1A5B	C2	5B	19 JNZ BACKT0 ;YES - LOCATE PREVIOUS FIELD	
7162	1A5E	3A	C1	FF LDA CURCOL ;NO - FIND PREVIOUS SET TAB	
7163	1A61	3D	.	. DCR A ;START AT PREVIOUS COLUMN	
7164	1A62	2E	BF	. MVI L,LFTMGN-BASE	
7165	1A64	BE	.	. CMP M ;WHERE IS CURSOR?	
7166	1A65	CA	C9	12 JZ CURPD4 ;AT MARGIN - SET DISPLAY	
7167	1A68	F2	92	1A JP BKT100 ;AFTER MARGIN - FIND PREV TA	
7168	1A6B	.	.	. ;	
7169	1A6B	.	.	. ; CURSOR AT BEGINNING OF LINE - LOCATE TAB IN	
7170	1A6B	.	.	. ; PREVIOUS LINE	
7171	1A6B	.	.	. ;	
7172	1A6B	3A	6B	FF LDA MLKROW ;GET MEMORY LOCK ROW	
7173	1A6E	2E	C0	. MVI L,CURROW	
7174	1A70	8E	.	. CMP M ;CURRENT ROW = LOCK ROW?	
7175	1A71	C2	7B	1A JNZ BKT010 ;NO - MOVE CURSOR UP ONE ROW	
7176	1A74	CD	CE	0C CALL ROLLDN ;YES - ROLL DOWN ONE LINE	
7177	1A77	C8	.	. RZ ;CAN'T ROLL DOWN - EXIT	
7178	1A78	C3	7F	1A JMP BKT050 ;GO LOCATE LAST TAB SET	
7179	1A7B	.	.	. ;	
7180	1A7B	.	.	. ; CURSOR NOT AT TOP OF FREE AREA - MOVE UP 1 LINE	
7181	1A7B	.	.	. ;	
7182	1A7B	.	.	. BKT010 EQU \$	
7183	1A7B	7E	.	. MOV A,M ;GET CURRENT ROW NUMBER	
7184	1A7C	B7	.	. ORA A ;ROW = 0	
7185	1A7D	C8	.	. RZ ;YES - DON'T BACK TAB WHEN	
7186	1A7E	.	.	. ; CURSOR IS LOCATED IN ROW	
7187	1A7E	.	.	. ; ZERO AND DISPLAY LOCK ON	
7188	1A7E	35	.	. DCR M ;NO - DECREMENT ROW NUMBER	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 220
=====
7190     1A7F      . . .      ;
7191     1A7F      . . .      ; PREVIOUS ROW LOCATED - LOCATE LAST TAB SET
7192     1A7F      . . .      ;
7193     1A7F      . . .      BKT050 EQU $
7194     1A7F      3A 81 FF      LDA HTBTBL+9 ;GET LAST TAB ENTRY
7195     1A82     E6 80 .      ANI 200Q      ;LAST TAB SET?
7196     1A84     3E 4F .      MVI A,MAXCOL ;(SET FOR LAST COLUMN-1)
7197     1A86     CA 91 1A    JZ BKT060     ;NO - LOCATE LAST TAB
7198     1A89     3C . .      INR A         ;YES - SET FOR LAST COLUMN #
7199     1A8A     4F . .      MOV C,A
7200     1A8B     2E BE .      MVI L,RHTMGN-BASE
7201     1A8D     BE . .      CMP M         ;RIGHT MARGIN = LAST COLUMN?
7202     1A8E     CA C9 12    JZ CURP04     ;YES - SET CURSOR TO LAST CO
7203     1A91     . . .      BKT060 EQU $ ;NO - SET TO MAXCOL-1 AND
7204     1A91     3D . .      DCR A         ;LOCATE PREVIOUS TAB
7205     1A92     . . .      ;
7206     1A92     . . .      ; LOCATE PREVIOUS TAB (A = CURRENT COLUMN - 1)
7207     1A92     . . .      ;
7208     1A92     . . .      BKT100 EQU $
7209     1A92     3C . .      INR A         ;RESTORE CURRENT COLUMN
7210     1A93     47 . .      MOV B,A       ;SAVE IT
7211     1A94     F6 07 .      ORI 7Q        ;SET TO COLUMN CORRESPONDING
7212     1A96     . . .      ; TO LAST BIT OF TAB BYTE
7213     1A96     2E BF .      MVI L,LFTMGN-BASE
7214     1A98     96 . .      SUB M         ;COMPUTE NUMBER OF CHARS
7215     1A99     4F . .      MOV C,A       ;TO SEARCH
7216     1A9A     78 . .      MOV A,B       ;RECALL CURRENT COLUMN
7217     1A9B     CD 3F 16    CALL FNDBT1   ;GET BYTE MASK AND
7218     1A9E     . . .      ; CORRESPONDING TABLE BYTE
7219     1A9E     3D . .      DCR A         ;SET FOR MASK TO MASK OFF
7220     1A9F     A6 . .      ANA M         ;SUCCEEDING TABS
7221     1AA0     . . .      ;
7222     1AA0     . . .      ; LOCATE PREVIOUS TAB SETTING
7223     1AA0     . . .      ;
7224     1AA0     . . .      BKT110 EQU $
7225     1AA0     06 08 .      MVI B,8       ;INITIALIZE BIT COUNT
7226     1AA2     . . .      BKT120 EQU $
7227     1AA2     07 . .      RLC           ;TAB SET?
7228     1AA3     D2 B4 1A    JNC BKT150    ;NO - BACK UP ANOTHER COLUMN
7229     1AA6     . . .      ;
7230     1AA6     . . .      ; TAB LOCATED - SET CURSOR (C = TAB COLUMN)
7231     1AA6     . . .      ;
7232     1AA6     . . .      BKT130 EQU $
7233     1AA6     5F . .      MOV E,A       ;SAVE A-REGISTER
7234     1AA7     79 . .      MOV A,C       ;PUT COLUMN NUMBER IN A-REG
7235     1AA8     E5 . .      PUSH H        ;SAVE H AND L
7236     1AA9     2A BE FF    LHLD RHTMGN   ;GET MARGIN SETTINGS
7237     1AAC     84 . .      ADD H         ;COMPUTE TAB COLUMN LOCATION
7238     1AAD     2C . .      INR L         ;IS TAB LOCATION BEYOND LEFT
7239     1AAE     BD . .      CMP L         ;MARGIN?
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 221
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 221
7240	1AAF	E1 . .	POP H ;(RESTORE H AND L)	
7241	1AB0	FA C9 12	JM CURP04 ;NO - LOCATE TAB AND RETURN	
7242	1AB3	7B . .	MOV A,E ;YES - RECALL A-REGISTER	

```
=====
```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 222
=====
7244     1AB4     . . .      ;*****
7245     1AB4     . . .      ; CONTINUE SCANNING BACKWARDS *
7246     1AB4     . . .      ;*****
7247     1AB4     . . .      BKT150 EQU $
7248     1AB4     0D . .     DCR C      ;COLUMN ZERO REACHED?
7249     1AB5     CA A6 1A    JZ BKT130  ;YES - SET CURSOR COLUMN
7250     1AB8     05 . .     DCR B      ;BYTE DONE?
7251     1AB9     C2 A2 1A    JNZ BKT120 ;NO - CONTINUE TO NEXT COLUM
7252     1ABC     2B . .     DCX H      ;YES - GET NEXT BYTE
7253     1ABD     7E . .     MOV A,M
7254     1ABE     C3 A0 1A    JMP BKT110 ;CHECK BYTE FOR TAB SET
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 223
7256	1AC1	.	.	;*****	
7257	1AC1	.	.	; CHAIN - SET H,L TO POINTER FROM MEMORY *	
7258	1AC1	.	.	;*****	
7259	1AC1	.	.	;	
7260	1AC1	.	.	; ENTRY: H,L = ADDRESS OF POINTER	
7261	1AC1	.	.	;	
7262	1AC1	.	.	; EXIT : A = LSB OF POINTER	
7263	1AC1	.	.	; H,L = POINTER VALUE	
7264	1AC1	.	.	;	
7265	1AC1	.	.	CHAIN0 EQU \$	
7266	1AC1	EB	.	XCHG ;PUT ADDRESS INTO H,L	
7267	1AC2	.	.	CHAIN1 EQU \$	
7268	1AC2	7D	.	MOV A,L ;COMPUTE LOCATION OF NEXT	
7269	1AC3	E6	F0	ANI 3770-BLKSM ;BLOCK POINTER IN BLOCK	
7270	1AC5	6F	.	MOV L,A ;GET THE NEXT BLOCK ADDRESS	
7271	1AC6	.	.	CHAIN EQU \$	
7272	1AC6	7E	.	MOV A,M ;GET LSB OF POINTER	
7273	1AC7	23	.	INX H	
7274	1AC8	66	.	MOV H,M ;PUT MSB INTO H-REGISTER	
7275	1AC9	6F	.	MOV L,A ;PUT LSB INTO L-REGISTER	
7276	1ACA	.	.	NOFNCT EQU \$;(NON-FUNCTION FOR ESC SEQ	
7277	1ACA	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 224
=====
7279     1ACB      . . .      ;*****
7280     1ACB      . . .      ; CHKFMS - CHECK FORMAT AND SOFT KEY DEFINE MODE *
7281     1ACB      . . .      ;*****
7282     1ACB      . . .      ;
7283     1ACB      . . .      ; ENTRY:  DON'T CARE
7284     1ACB      . . .      ;
7285     1ACB      . . .      ; EXIT :  Z - NEITHER MODE ENABLED
7286     1ACB      . . .      ;           A = 0
7287     1ACB      . . .      ;           NZ - MODE ENABLED
7288     1ACB      . . .      ;           A = -1, SOFT KEY MODE ENABLED
7289     1ACB      . . .      ;           A > 0, FORMAT MODE ONLY ENABLED
7290     1ACB      . . .      ;
7291     1ACB      . . .      CHKFMO EQU $
7292     1ACB      2E 6C      MVI L,SPOWL ;TURN OF SPOW LATCH FIRST
7293     1ACD      36 FF      MVI M,SPOWOF
7294     1ACF      . . .      CHKFMS EQU $
7295     1ACF      3A AE FF   LDA DSPTYP ;GET DISPLAY TYPE FLAG
7296     1AD2      B7 . .     ORA A ;SOFT KEY DISPLAY ON?
7297     1AD3      C0 . .     RNZ ;YES - RETURN
7298     1AD4      . . .      CHKFMT EQU $
7299     1AD4      3A F4 FF   LDA MOFLG1 ;NO - GET MODE FLAGS
7300     1AD7      E6 08      ANI FORMAT ;MASK FOR FORMAT FLAG AND
7301     1AD9      C9 . .     RET ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 225
7303	1ADA	.	.	*****	
7304	1ADA	.	.	; CHKMLK - CHECK FOR MEMORY LOCK ENABLED *	
7305	1ADA	.	.	*****	
7306	1ADA	.	.	;	
7307	1ADA	.	.	; ENTRY: DON'T CARE	
7308	1ADA	.	.	;	
7309	1ADA	.	.	; EXIT : Z => MEMORY LOCK ENABLED	
7310	1ADA	.	.	; NZ => MEMORY LOCK NOT ENABLED	
7311	1ADA	.	.	; A,H,L DESTROYED	
7312	1ADA	.	.	;	
7313	1ADA	.	.	CHKMLK EQU \$	
7314	1ADA	3A	F4 FF	LDA MDFLG1 ;GET SOFT MODE FLAGS	
7315	1ADD	2F	.	CMA ;MEMORY LOCK ENABLED FOR FUL	
7316	1ADE	E6	04 .	ANI MEMLOK ;LOCKOUT IF MEMORY LOCK SE	
7317	1AE0	21	6B FF	LXI H,MLKROW ;AND MEMORY LOCK ROW = 0	
7318	1AE3	86	.	ORA M	
7319	1AE4	C9	.	RET ;RETURN	
7320	1AE5	.	.	*****	
7321	1AE5	.	.	; CHKSFK - CHECK FOR SOFT KEY MODE *	
7322	1AE5	.	.	*****	
7323	1AE5	.	.	;	
7324	1AE5	.	.	; EXIT : Z - NORMAL MODE	
7325	1AE5	.	.	; A = 0	
7326	1AE5	.	.	; NZ - SOFT KEY DEFINE MODE	
7327	1AE5	.	.	; A DESTROYED	
7328	1AE5	.	.	;	
7329	1AE5	.	.	CHKSFK EQU \$	
7330	1AE5	3A	AE FF	LDA DSPTYP ;GET DISPLAY TYPE FLAG	
7331	1AE8	B7	.	ORA A ;SET Z FALSE IF SOFT KEY	
7332	1AE9	C9	.	RET ;ON AND RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 226
7334	1AEA	.	.	. ;*****	
7335	1AEA	.	.	. ; CD - CHARACTER DELETE *	
7336	1AEA	.	.	. ;*****	
7337	1AEA	.	.	. DELWRP EQU \$;DELETE WITH WRAP AROUND	
7338	1AEA	CD	CF	1A CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE	
7339	1AED	3E	20	. MVI A,WRPDEL ;(PUT WRAP FLAG IN A-REG)	
7340	1AEF	CC	94	18 CZ SETMF2 ;NO - SET WRAP AROUND FLAG	
7341	1AF2	.	.	. ;	
7342	1AF2	.	.	. CHRDEL EQU \$	
7343	1AF2	CD	E5	1A CALL CHKSFK ;SOFT KEY DEFINE MODE?	
7344	1AF5	CA	FD	1A JZ CHD010 ;NO - DU DELETE	
7345	1AF8	3A	C0	FF LDA CURROW ;YES - GET CURSOR ROW	
7346	1AFB	0F	.	. RRC ;IN DATA LINE?	
7347	1AFC	D0	.	. RNC ;NO - RETURN	
7348	1AFD	.	.	. CHD010 EQU \$;YES - DO DELETE	
7349	1AFD	AF	.	. XRA A ;ZERO SAVE AREA	
7350	1AFE	32	98	FF STA CHSAV	
7351	1B01	CD	63	1B CALL CHD000 ;DELETE A CHARACTER	
7352	1B04	3A	98	FF LDA CHSAV ;RECALL THE DELETED CHARACTE	
7353	1B07	B7	.	. URA A ;WAS IT A DISPLAY CONTROL?	
7354	1B08	FA	F2	1A JM CHRDEL ;YES - CONTINUE DELETING	
7355	1B0B	.	.	. ;*****	
7356	1B0B	.	.	. ; ADJUST FOR CHARACTERS BEYOND RIGHT MARGIN *	
7357	1B0B	.	.	. ;*****	
7358	1B0B	21	C1	FF LXI H,CURCOL	
7359	1B0E	3A	BE	FF LDA RHTMGN	
7360	1B11	BE	.	. CMP M ;CURSOR BEYOND RIGHT MARGIN?	
7361	1B12	D8	.	. RC ;YES - DON'T CHECK WRAP	
7362	1B13	46	.	. MOV B,M ;NO - SAVE CURRENT COLUMN	
7363	1B14	77	.	. MOV M,A ;SET COLUMN TO RIGHT MARGIN	
7364	1B15	57	.	. MOV D,A ;SAVE RIGHT MARGIN VALUE	
7365	1B16	2E	F4	. MVI L,MDFLG1-BASE	
7366	1B18	4E	.	. MOV C,M ;SAVE SOFT MODE FLAGS STATE	
7367	1B19	C5	.	. PUSH B ;AND CURRENT COLUMN	
7368	1B1A	79	.	. MOV A,C ;FORCE THE INSERT CHARACTER	
7369	1B1B	E6	FD	. ANI 377Q-INSCHR ;MODE OFF	
7370	1B1D	77	.	. MOV M,A	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 227
7372	1B1E	.	.	*****	
7373	1B1E	.	.	; DELETE PERFORMED - CHECK FOR WRAP AROUND *	
7374	1B1E	.	.	*****	
7375	1B1E	21	6F FF	LXI H,MFLGS2 ;GET TERMINAL MODE FLAGS	
7376	1B21	7E	.	MOV A,M ;MASK OUT DELETE WRAP FLAG	
7377	1B22	E6	DF .	ANI 377Q-WRPDEL	
7378	1B24	BE	.	CMP M ;DELETE WRAP AROUND ENABLED?	
7379	1B25	CA	5A 1B	JZ CHD050 ;NO - EXIT	
7380	1B28	77	.	MOV M,A ;YES - UPDATE MODE FLAGS	
7381	1B29	.	.	*****	
7382	1B29	.	.	; TRANSFER A CHARACTER UP FROM THE NEXT LINE *	
7383	1B29	.	.	*****	
7384	1B29	.	.	CHD020 EQU \$	
7385	1B29	3E	20 .	MVI A,ABLNK ;PRESET DELETED CHARACTER	
7386	1B2B	32	98 FF	STA CHSAV ;TO A BLANK	
7387	1B2E	21	C0 FF	LXI H,CURROW ;SET TO DELETE FIRST	
7388	1B31	34	.	INR M ;CHARACTER AT LEFT MARGIN	
7389	1B32	23	.	INX H ;FROM NEXT ROW	
7390	1B33	3A	BF FF	LDA LFTMGN	
7391	1B36	77	.	MOV M,A	
7392	1B37	CD	CA 07	CALL RCADR4 ;CHARACTER EXIST?	
7393	1B3A	CC	72 1B	CZ CHRDL1 ;YES - DELETE IT	
7394	1B30	21	C0 FF	LXI H,CURROW ;RESTORE ROW NUMBER AND SET	
7395	1B40	35	.	DCR M ;COLUMN TO RIGHT MARGIN	
7396	1B41	23	.	INX H	
7397	1B42	3A	BE FF	LDA RHTMGN	
7398	1B45	77	.	MOV M,A	
7399	1B46	3A	98 FF	LDA CHSAV ;GET THE DELETED CHARACTER	
7400	1B49	FE	20 .	CPI ABLNK ;BLANK CHARACTER DELETED?	
7401	1B4B	CA	5A 1B	JZ CHD050 ;YES - EXIT	
7402	1B4E	06	00 .	MVI B,0 ;NO - SET TO FORCE ENHANCE	
7403	1B50	CD	45 24	CALL DISPC2 ;DISPLAY THE CHARACTER	
7404	1B53	3A	98 FF	LDA CHSAV ;RECALL THE DELETED CHARACTE	
7405	1B56	B7	.	ORA A ;WAS IT ASCII?	
7406	1B57	FA	29 1B	JM CHD020 ;NO - TRANSFER ANOTHER BYTE	
7407	1B5A	.	.	*****	
7408	1B5A	.	.	; EXIT - RESTORE CURSOR COLUMN AND "MDFLG1" *	
7409	1B5A	.	.	*****	
7410	1B5A	.	.	CHD050 EQU \$	
7411	1B5A	C1	.	POP B ;RECALL ORIGINAL VALUES	
7412	1B5B	21	C1 FF	LXI H,CURCOL	
7413	1B5E	70	.	MOV M,B ;RESTORE CURSOR COLUMN	
7414	1B5F	2E	F4 .	MVI L,MDFLG1-BASE	
7415	1B61	71	.	MOV M,C ;RESTORE "MDFLG1"	
7416	1B62	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 228
=====
7418     1863      .      .      .      CHD000 EQU $
7419     1863      CD     CA     07      CALL RCADR4      ;DOES CHARACTER EXIST?
7420     1866      C0      .      .      RNZ              ;NO - RETURN
7421     1867      CD     F3     1B      CALL CHD500      ;SKIP OVER SINGLE DISPLAY
7422     186A      .      .      .      ;              ENHANCEMENT CODE
7423     186A      CD     D4     1A      CALL CHKFMT      ;FORMAT MODE?
7424     186D      CA     72     1B      JZ   CHD100      ;NO - DELETE THE CHARACTER
7425     1870      04      .      .      INR   B          ;CURSOR IN PROTECTED FIELD?
7426     1871      C8      .      .      RZ              ;YES - RETURN
7427     1872      .      .      .      ;*****
7428     1872      .      .      .      ; CHRDL1 - DELETE ONE CHARACTER *
7429     1872      .      .      .      ;*****
7430     1872      .      .      .      ;
7431     1872      .      .      .      ; ENTRY:  C = CHARACTER COLUMN POSITION
7432     1872      .      .      .      ;          D,E = ADDRESS OF CHAR TO BE DELETED
7433     1872      .      .      .      ;
7434     1872      .      .      .      ; EXIT :  ALL REGISTERS DESTROYED
7435     1872      .      .      .      ;          CHSAV = CHARACTER DELETED (UNCHANGED
7436     1872      .      .      .      ;          IF A CHARACTER HAS NOT BEEN DELETED)
7437     1872      .      .      .      ;
7438     1872      .      .      .      CHRDL1 EQU $
7439     1872      .      .      .      CHD100 EQU $
7440     1872      1A      .      .      LDAX D          ;GET CHARACTER TO BE DELETED
7441     1873      FE     CC     .      CPI EOL         ;IS IT EOL?
7442     1875      C8      .      .      RZ              ;YES - RETURN
7443     1876      32     98     FF      STA CHSAV       ;SAVE THE DELETED CHARACTER
7444     1879      62      .      .      MOV  H,D        ;H,L = ADDR OF CHAR TO FILL
7445     187A      6B      .      .      MOV  L,E        ;D,E = ADDR OF CHAR TO MOVE
7446     187B      .      .      .      ;*****
7447     187B      .      .      .      ; MOVE CHARACTERS DOWN TO PREVIOUS POSITION *
7448     187B      .      .      .      ;*****
7449     187B      .      .      .      CHD110 EQU $
7450     187B      CD     90     0C      CALL NXTCHR     ;GET THE NEXT CHARACTER
7451     187E      C2     CD     1B      JNZ CHD210      ;EOL LINK - TERMINATE DELETE
7452     1881      47      .      .      MOV  B,A        ;SAVE CHARACTER IN B-REGISTE
7453     1882      FE     C0     .      CPI ENHLIM+1    ;ASCII OR ENHANCEMENT CODE?
7454     1884      DA     A2     1B      JC   CHD120     ;YES - SEE IF PAST MARGIN
7455     1887      .      .      .      ;*****
7456     1887      .      .      .      ; FORMAT CONTROL CODE FOUND - CHECK FUNCTION *
7457     1887      .      .      .      ;*****
7458     1887      FE     CC     .      CPI EOL         ;END OF LINE?
7459     1889      CA     E1     1B      JZ   CHD250     ;YES - TERMINATE DELETE
7460     188C      FE     C3     .      CPI FILL        ;END OF LINE FILL?
7461     188E      CA     C9     1B      JZ   CHD200     ;YES - TERMINATE DELETE
7462     1891      CD     EA     1B      CALL CHD400     ;FORMAT MODE & DELETE ASCII?
7463     1894      CA     A2     1B      JZ   CHD120     ;NO - MOVE NEW CHARACTER
7464     1897      78      .      .      MOV  A,B        ;YES - GET CHAR TO BE MOVED
7465     1898      FE     C0     .      CPI STPR        ;IS IT START PROTECT?
7466     189A      CA     F6     1D      JZ   CLER02     ;YES - CLEAR REST OF FIELD
7467     189D      .      .      .      ;              AND TERMINATE DELETE
=====

```

13255
2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 229
=====
7468     189D     FE C5      .              CPI ALPHA      ;TYPE DEFINITION?
7469     189F     F2 7B 1B      JP CHD110     ;YES - SKIP OVER CHARACTER
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 230
7471	1BA2	.	.	. ;*****	
7472	1BA2	.	.	. ; CHARACTER TO BE MOVED - CHECK MARGIN *	
7473	1BA2	.	.	. ;*****	
7474	1BA2	.	.	. CHD120 EQU \$	
7475	1BA2	3A	BE	FF LDA RHTMGN	
7476	1BA5	B9	.	. CMP C ;CHAR FROM BEYOND MARGIN?	
7477	1BA6	C2	B0	1B JNZ CHD130 ;NO - CONTINUE DELETE	
7478	1BA9	3A	98	FF LDA CHSAV ;YES - GET DELETED CHARACTER	
7479	1BAC	B7	.	. ORA A ;IS IT ASCII?	
7480	1BAD	36	20	. MVI M,ABLNK ;(SET BLANK BY DEFAULT)	
7481	1BAF	F0	.	. RP ;YES - TERMINATE DELETE	
7482	1BB0	.	.	. ; NO - PUT CHAR INTO PREV CHAR	
7483	1BB0	.	.	. ;*****	
7484	1BB0	.	.	. ; MOVE CHARACTER INTO PREVIOUS CHARACTER POSITON *	
7485	1BB0	.	.	. ;*****	
7486	1BB0	.	.	. CHD130 EQU \$	
7487	1BB0	70	.	. MOV M,B ;REPLACE PREVIOUS CHARACTER	
7488	1BB1	78	.	. MOV A,B	
7489	1BB2	B7	.	. ORA A ;IS CHARACTER ASCII?	
7490	1BB3	FA	B7	1B JM CHD140 ;NO - ADVANCE TO NEXT CHAR	
7491	1BB6	0C	.	. INR C ;YES - INCREMENT COLUMN #	
7492	1BB7	.	.	. CHD140 EQU \$	
7493	1BB7	CD	8F	0C CALL NXTCH0 ;GET THE NEXT CHARACTER	
7494	1BBA	FE	C5	. CPI ALPHA ;TYPE DEFINITION?	
7495	1BBC	DA	C5	1B JC CHD150 ;NO - CONTINUE MOVING CHARS	
7496	1BBF	CD	EA	1B CALL CHD400 ;FORMAT MODE & DELETE ASCII?	
7497	1BC2	C4	90	0C CNZ NXTCHR ;YES - ADVANCE TO NEXT CHAR	
7498	1BC5	.	.	. CHD150 EQU \$	
7499	1BC5	EB	.	. XCHG ;RESTORE REGISTER POSITIONS	
7500	1BC6	C3	78	1B JMP CHD110 ;MOVE NEXT CHARACTER	

				=====		PAGE 231	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS			
=====							
7502	1BC9	.	.	.	;*****		
7503	1BC9	.	.	.	; END OF LINE FILL CHARACTER FOUND - CLEAR THE *		
7504	1BC9	.	.	.	; REST OF THE LINE *		
7505	1BC9	.	.	.	;*****		
7506	1BC9	.	.	.	CHD200 EQU \$		
7507	1BC9	CD	C1	1A	CALL CHAIN0	;GET END OF LINE LINK IN H,L	
7508	1BCC	EB	.	.	XCHG	;EXCHANGE H,L AND D,E	
7509	1BCD	.	.	.	;*****		
7510	1BCD	.	.	.	; END OF LINE LINK FOUND - CLEAR THE REST OF *		
7511	1BCD	.	.	.	; THE LINE *		
7512	1BCD	.	.	.	;*****		
7513	1BCD	.	.	.	CHD210 EQU \$		
7514	1BCD	CD	EA	1B	CALL CHD400	;FORMAT MODE & DELETE ASCII?	
7515	1BD0	EB	.	.	XCHG	;(SET D,E TO LAST CHAR ADD	
7516	1BD1	2B	.	.	DCX H	;H,L TO LSB OF NEXT LINE	
7517	1BD2	.	.	.		LINK)	
7518	1BD2	C2	E5	1B	JNZ CHD260	;YES - CLEAR REST OF FIELD	
7519	1BD5	.	.	.		TO LSB OF NEXT LINE LINK	
7520	1BD5	3A	98	FF	LDA CHSAV	;RECALL DELETED CHARACTER	
7521	1BD8	B7	.	.	ORA A	;WAS IT ASCII?	
7522	1BD9	F2	E5	1B	JP CHD260	;YES - END LINE WITH EOL	
7523	1BDC	3E	C3	.	MVI A,FILL	;NO - END LINE WITH FILL	
7524	1BD0E	C3	BA	1D	JMP CLERL1	;CLEAR REST OF LINE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 232
=====
7526     1BE1      . . .      ;*****
7527     1BE1      . . .      ; EOL FOUND - CLEAR THE REST OF THE LINE *
7528     1BE1      . . .      ;*****
7529     1BE1      . . .      CHD250 EQU $
7530     1BE1      CD C1 1A    CALL CHAIN0      ;GET EOL LINK IN H,L
7531     1BE4      2B . .     DCX H            ;SET TO LSB OF NEXT LINE LIN
7532     1BE5      . . .      CHD260 EQU $      ;CLEAR THE REST OF THE LINE
7533     1BE5      3E CC .    MVI A,EOL        ;TERMINATING WITH AN EOL
7534     1BE7      C3 BA 1D    JMP CLERL1
7535     1BEA      . . .      ;*****
7536     1BEA      . . .      ; CHD400 - CHECK FOR FORMAT MODE ENABLED AND *
7537     1BEA      . . .      ; DISPLAYABLE ASCII CHARACTER DELETED *
7538     1BEA      . . .      ;*****
7539     1BEA      . . .      ;
7540     1BEA      . . .      ; EXIT :  NZ - FORMAT MODE AND DELETE ASCII
7541     1BEA      . . .      ;          Z - NOT FORMAT MODE OR NON-DISPLAY
7542     1BEA      . . .      ;          CODE DELETED
7543     1BEA      . . .      ;
7544     1BEA      . . .      CHD400 EQU $
7545     1BEA      3A 98 FF    LDA CHSAV        ;GET CHARACTER DELETED
7546     1BED      B7 . .     ORA A            ;IS IT DISPAYABLE ASCII
7547     1BEE      F2 D4 1A    JP  CHKFM1       ;YES - CHECK FOR FORMAT MODE
7548     1BF1      AF . .     XRA A            ;NO - RETURN Z
7549     1BF2      C9 . .     RET
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 233
7551	1BF3	.	.	*****	
7552	1BF3	.	.	; CHD500 - CHECK FOR DISPLAY ENHANCEMENT DELETE *	
7553	1BF3	.	.	*****	
7554	1BF3	.	.	;	
7555	1BF3	.	.	; ENTRY: D,E = CHARACTER TO BE DELETED	
7556	1BF3	.	.	;	
7557	1BF3	.	.	; EXIT : D,E = ACTUAL CHARACTER TO DELETE	
7558	1BF3	.	.	; A,L DESTROYED	
7559	1BF3	.	.	;	
7560	1BF3	.	.	CHD500 EQU \$	
7561	1BF3	1A	.	LDAX D ;GET CHARACTER TO BE DELETED	
7562	1BF4	87	.	ADD A ;DISPLAY ENHANCEMENT CODE?	
7563	1BF5	D0	.	RNC ;ASCII - LET IT BE DELETED	
7564	1BF6	F8	.	RM ;FORMAT CONTROL - DELETE IT	
7565	1BF7	2E	02	MVI L,2 ;YES - LOOK FOR POSSIBLE	
7566	1BF9	D5	.	PUSH D ;DOUBLE ENHANCEMENT CODE	
7567	1BFA	.	.	CHD510 EQU \$	
7568	1BFA	CD	90	CALL NXTCHR ;GET THE NEXT CHARACTER	
7569	1BFD	C2	07	JNZ CHD515 ;EXIT IF EOL LINK	
7570	1C00	87	.	ADD A ;ENHANCEMENT CODE?	
7571	1C01	D2	09	JNC CHD520 ;ASCII - CHECK FOR SCAN DONE	
7572	1C04	FA	FA	JM CHD510 ;FORMAT CONTROL - CONTINUE	
7573	1C07	.	.	CHD515 EQU \$	
7574	1C07	D1	.	POP D ;YES - DELETE ENHANCEMENT	
7575	1C08	C9	.	RET	
7576	1C09	.	.	*****	
7577	1C09	.	.	; ASCII CHARACTER FOUND - CHECK FOR SCAN DONE *	
7578	1C09	.	.	*****	
7579	1C09	.	.	CHD520 EQU \$	
7580	1C09	2D	.	DCR L ;NEXT ASCII CHARACTER FOUND?	
7581	1C0A	C2	FA	JNZ CHD510 ;NO - CONTINUE SCAN	
7582	1C0D	.	.	*****	
7583	1C0D	.	.	; NEXT ASCII CHARACTER OR EOL LINK FOUND - *	
7584	1C0D	.	.	; DON'T DELETE DISPLAY ENHANCEMENT CODE *	
7585	1C0D	.	.	*****	
7586	1C0D	D1	.	POP D ;RECALL ORIGINAL ADDRESS	
7587	1C0E	C3	90	JMP NXTCHR ;SET TO DELETE NEXT CHAR	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 234
=====
7589     1C11      . . .      ;*****
7590     1C11      . . .      ; CHRDL2 - DELETE CHARACTER W/REGISTER SAVE *
7591     1C11      . . .      ;*****
7592     1C11      . . .      ;
7593     1C11      . . .      ; ENTRY: C = CHARACTER COLUMN POSITION
7594     1C11      . . .      ;          D,E = ADDRESS OF CHAR TO BE DELETED
7595     1C11      . . .      ;
7596     1C11      . . .      ; EXIT : B,C = B,C(ENTRY)
7597     1C11      . . .      ;          D,E = D,E(ENTRY) + 1
7598     1C11      . . .      ;          A,H,L DESTROYED
7599     1C11      . . .      ;
7600     1C11      . . .      CHRDL2 EQU $
7601     1C11      C5 . .      PUSH B          ;SAVE REGISTERS B,C
7602     1C12      D5 . .      PUSH D          ;AND D,E
7603     1C13      CD 72 18    CALL CHRDL1     ;DELETE THE CHARACTER
7604     1C16      D1 . .      POP D           ;RESTORE REGISTER D,E
7605     1C17      C1 . .      POP B          ;AND B,C
7606     1C18      13 . .      INX D          ;INCREMENT D,E
7607     1C19      C9 . .      RET            ;RETURN
=====
  
```


				PAGE 235	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	
7609	1C1A	.	.	.	;*****
7610	1C1A	.	.	.	; CHRINS - INSERT CHARACTER *
7611	1C1A	.	.	.	;*****
7612	1C1A	.	.	.	;
7613	1C1A	.	.	.	; ENTRY: A = CHARACTER TO BE INSERTED
7614	1C1A	.	.	.	; CURROW,CURCOL = DISPLAY POSITION WHERE
7615	1C1A	.	.	.	; INSERT IS TO BE DONE
7616	1C1A	.	.	.	;
7617	1C1A	.	.	.	; EXIT : A = 0, INSERT NOT DONE
7618	1C1A	.	.	.	; A # 0, INSERT PERFORMED
7619	1C1A	.	.	.	; DCHAR DESTROYED
7620	1C1A	.	.	.	; B-L DESTROYED
7621	1C1A	.	.	.	;
7622	1C1A	.	.	.	; CHARACTER IS INSERTED IMMEDIATELY AHEAD OF THE
7623	1C1A	.	.	.	; CHARACTERS LOCATED AT THE SPECIFIED ROW AND
7624	1C1A	.	.	.	; COLUMN POSITIONS
7625	1C1A	.	.	.	;
7626	1C1A	.	.	.	CHRINS EQU \$
7627	1C1A	32	89	FF	STA DCHAR ;SAVE CHAR TO BE INSERTED
7628	1C1D	3E	FF	.	MVI A,3770 ;INHIBIT LINE EXTENSION
7629	1C1F	32	91	FF	STA BLKFI
7630	1C22	CD	CA	07	CALL RCADR4 ;DOES DISPLAY POSITION EXIST
7631	1C25	C2	4C	24	JNZ DISPLA ;NO - TRY TO EXTEND LINE
7632	1C28	.	.	.	; YES - INSERT THE CHARACTER

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 236
7634	1C28	.	.	*****	
7635	1C28	.	.	; CRI100 - ENTRY TO STORE CHARACTER FOR INSERT *	
7636	1C28	.	.	; CHARACTER MODE *	
7637	1C28	.	.	*****	
7638	1C28	.	.	;	
7639	1C28	.	.	; ENTRY: C = COLUMN NUMBER	
7640	1C28	.	.	; D,E = ADDR WHERE INSERT IS TO BE MADE	
7641	1C28	.	.	; H = BASEH	
7642	1C28	.	.	;	
7643	1C28	.	.	CRI100 EQU \$	
7644	1C28	CD	90 11	CALL CKPROT ;CURSOR IN PROTECTED FIELD?	
7645	1C28	CA	3D 25	JZ DIS092 ;YES - TAB TO NEXT FIELD	
7646	1C2E	.	.	CRI104 EQU \$	
7647	1C2E	2E	89 .	MVI L,DCHAR-BASE	
7648	1C30	46	.	MOV B,M ;GET CHAR TO BE INSERTED	
7649	1C31	EB	.	XCHG ;PUT CHAR ADDRESS INTO H,L	
7650	1C32	.	.	CRI110 EQU \$	
7651	1C32	78	.	MOV A,B ;IS THIS CONTROL CODE?	
7652	1C33	87	.	ORA A	
7653	1C34	FA	38 1C	JM CRI120 ;YES - DON'T COUNT COLUMN	
7654	1C37	0C	.	INR C ;INCREMENT COLUMN	
7655	1C38	.	.	CRI120 EQU \$	
7656	1C38	7E	.	MOV A,M ;GET CHAR IN CURRENT ADDR	
7657	1C39	70	.	MOV M,B ;STORE NEW CHAR	
7658	1C3A	47	.	MOV B,A ;SAVE OLD CHAR IN B	
7659	1C3B	2B	.	DCX H ;MOVE TO NEXT CHARACTER	
7660	1C3C	3E	50 .	MVI A,MAXCOL+1	
7661	1C3E	89	.	CMP C ;STORE DONE AT END OF LINE?	
7662	1C3F	CA	EA 1C	JZ CRI305 ;YES - TERMINATE INSERT	
7663	1C42	3A	BE FF	LDA RHTMGN ;GET RIGHT MARGIN COLUMN	
7664	1C45	3C	.	INR A ;WAS THE LAST STORE DONE	
7665	1C46	B9	.	CMP C ;AT THE RIGHT MARGIN?	
7666	1C47	CA	E2 1C	JZ CRI300 ;YES - TERMINATE INSERT	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 237
7668	1C4A	.	.	*****	
7669	1C4A	.	.	; PROCESS NEXT CHARACTER OF BLOCK *	
7670	1C4A	.	.	*****	
7671	1C4A	.	.	CRI140 EQU \$	
7672	1C4A	7E	.	MOV A,M ;GET THE NEXT CHARACTER	
7673	1C4B	FE	C0	CPI ENHLIM+1 ;ASCII OR DISPLAY CONTROL?	
7674	1C4D	DA	32 1C	JC CRI110 ;YES - MOVE THE BYTE	
7675	1C50	FE	D0	CPI LNKLIM ;IS IT A LINK BYTE?	
7676	1C52	D2	B6 1C	JNC CRI200 ;YES - MOVE TO NEXT BLOCK	
7677	1C55	FE	CC	CPI EOL ;IS IT END OF LINE?	
7678	1C57	CA	92 1C	JZ CRI158 ;YES - ADD LAST CHAR TO LINE	
7679	1C5A	FE	C3	CPI FILL ;END OF LINE FILL CHARACTER?	
7680	1C5C	CA	9D 1C	JZ CRI159 ;YES - ADD BYTE TO END	
7681	1C5F	3A	89 FF	LDA DCHAR ;NO - FIELD CHECK CHARACTER	
7682	1C62	B7	.	ORA A ;IS ADDED CHARACTER ASCII?	
7683	1C63	FA	32 1C	JM CRI110 ;NO - CONTINUE INSERT MOVE	
7684	1C66	CD	CF 1A	CALL CHKFMS ;FORMAT MODE ENABLED?	
7685	1C69	CA	32 1C	JZ CRI110 ;NO - CONTINUE INSERT	
7686	1C6C	7E	.	MOV A,M ;YES - RECALL THE BYTE	
7687	1C6D	FE	C0	CPI STPR ;IS CHARACTER A START PROT?	
7688	1C6F	CA	7B 1C	JZ CRI150 ;YES - CHECK INSERT TYPE	
7689	1C72	FE	C5	CPI ALPHA ;FIELD TYPE DEFINITION?	
7690	1C74	FA	32 1C	JM CRI110 ;NO - CONTINUE INSERT	
7691	1C77	2B	.	DCX H ;YES - ADVANCE TO NEXT BYTE	
7692	1C78	C3	4A 1C	JMP CRI140 ;LOOK TO NEXT CHARACTER	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 238
=====
7694     1C7B      . . .      ;*****
7695     1C7B      . . .      ; END OF FIELD - ASCII CODE INSERTED *
7696     1C7B      . . .      ;*****
7697     1C7B      . . .      CRI150 EQU $
7698     1C7B      78 . .     MOV A,B      ;GET CHAR WHICH ROLLED OFF
7699     1C7C      B7 . .     ORA A        ;IS IT ASCII?
7700     1C7D      F2 88 1C   JP CRI154    ;YES - DELETE PREV CONTROLS
7701     1C80      . . .      CRI152 EQU $ ;NO - BACK UP ANOTHER CHAR
7702     1C80      CD 87 1D   CALL CRI500  ;IS PREVIOUS CHARACTER ASCII
7703     1C83      FA 80 1C   JM CRI152    ;NO - CONTINUE BACKING UP
7704     1C86      36 80 .    MVI M,200Q   ;YES - TEMPORARILY REPLACE
7705     1C88      . . .      ;          ASCII WITH DUMMY CONTROL
7706     1C88      . . .      CRI154 EQU $
7707     1C88      CD 87 1D   CALL CRI500  ;PREVIOUS CHARACTER ASCII?
7708     1C8B      FA 88 1C   JM CRI154    ;NO - CONTINUE BACKING UP
7709     1C8E      2B . .     DCX H        ;MOVE TO NEXT CHARACTER
7710     1C8F      C3 F6 1D   JMP CLER02   ;CLEAR REST OF FIELD
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 239
7712	1C92	.	.	*****	
7713	1C92	.	.	; EOL FOUND *	
7714	1C92	.	.	; ADD LAST CHARACTER TO LINE *	
7715	1C92	.	.	*****	
7716	1C92	.	.	CRI158 EQU \$	
7717	1C92	78	.	MOV A,B ;GET CHARACTER	
7718	1C93	B7	.	ORA A ;IS THIS CONTROL CHAR?	
7719	1C94	FA	A0 1C	JM CRI160 ;YES - ADD CHAR	
7720	1C97	79	.	MOV A,C ;NO - CHAR IS ASCII	
7721	1C98	FE	4F .	CPI MAXCOL ;IS THIS MAX COLUMN?	
7722	1C9A	C2	A1 1C	JNZ CRI170 ;NO - ADD CHAR	
7723	1C9D	.	.	CRI159 EQU \$	
7724	1C9D	70	.	MOV M,B ;ASCII CHARACTER INSERTED TO	
7725	1C9E	B7	.	ORA A ;MAXIMUM COLUMN - OVERLAY	
7726	1C9F	C9	.	RET ;EOL AND RETURN NZ	
7727	1CA0	.	.	*****	
7728	1CA0	.	.	; EOL CANNOT BE OVERLAYED *	
7729	1CA0	.	.	; ADD NEW CHAR TO LINE *	
7730	1CA0	.	.	*****	
7731	1CA0	.	.	CRI160 EQU \$	
7732	1CA0	0D	.	DCR C	
7733	1CA1	.	.	CRI170 EQU \$	
7734	1CA1	EB	.	XCHG ;PUT H,L INTO D,E	
7735	1CA2	.	.	CRI180 EQU \$	
7736	1CA2	21	89 FF	LXI H,DCHAR ;SAVE CHARACTER TO BE ADDED	
7737	1CA5	70	.	MOV M,B	
7738	1CA6	2E	C1 .	MVI L,CURCOL-BASE	
7739	1CA8	46	.	MOV B,M ;GET CURRENT CURSOR COLUMN	
7740	1CA9	C5	.	PUSH B ;AND SAVE IT	
7741	1CAA	71	.	MOV M,C ;SET "CURCOL" TO INSERT COL	
7742	1CAB	0E	00 .	MVI C,0 ;SET # OF CHARS NEEDED TO 1	
7743	1CAD	.	.	; (VALUE IN C IS ONE LESS)	
7744	1CAD	CD	AB 09	CALL DISPL1 ;BUILD NECESSARY BLOCKS	
7745	1CB0	C1	.	POP B ;RESTORE ORIGINAL CURSOR	
7746	1CB1	21	C1 FF	LXI H,CURCOL ;COLUMN NUMBER	
7747	1CB4	70	.	MOV M,B	
7748	1CB5	C9	.	RET ;RETURN (A=MEMORY LOCK STATE)	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 240
=====
7750      1CB6      .      .      .      ;*****
7751      1CB6      .      .      .      ; LINK FOUND - MOVE TO NEXT BLOCK *
7752      1CB6      .      .      .      ;*****
7753      1CB6      .      .      .      CRI200 EQU $
7754      1CB6      22     96     FF      SHLD LNKSAV      ;SAVE CURRENT BLOCK ADDRESS
7755      1CB9      28     .      .      DCX H            ;GET THE LSB OF THE LINK
7756      1CBA      7E     .      .      MOV A,M
7757      1CB8      2F     .      .      CMA              ;IS IT AN EOL LINK (LOWER
7758      1CBC      E6     0F     .      ANI BLKSM        ;FOUR BITS NOT ALL ONES)?
7759      1CBE      C2     C7     1C      JNZ CRI240       ;YES - EXTEND THE LINE
7760      1CC1      CD     C6     1A      CALL CHAIN       ;NO - GET NEXT BLOCK ADDRESS
7761      1CC4      C3     4A     1C      JMP CRI140       ;CONTINUE INSERT CHARACTER
7762      1CC7      .      .      .      ;*****
7763      1CC7      .      .      .      ; NEW BLOCK REQUIRED *
7764      1CC7      .      .      .      ;*****
7765      1CC7      .      .      .      CRI240 EQU $
7766      1CC7      78     .      .      MOV A,B          ;SAVE CHARACTER BEING MOVED
7767      1CC8      32     9D     FF      STA TEMP
7768      1CCB      23     .      .      INX H            ;GET THE LAST CHARACTER OF
7769      1CCC      23     .      .      INX H            ;THE CURRENT BLOCK TO BE
7770      1CCD      46     .      .      MOV B,M          ;STORED AGAIN IN THE SAME
7771      1CCE      EB     .      .      XCHG             ;LOCATION BY "DISPL1"
7772      1CCF      00     .      .      DCK C            ;GET COLUMN # OF PREV CHAR
7773      1CD0      CD     A2     1C      CALL CRI180      ;ADD BLOCK
7774      1CD3      B7     .      .      ORA A            ;IS MEMORY LOCKED?
7775      1CD4      CA     DC     1C      JZ CRI260        ;YES - BLOCK NOT ADDED
7776      1CD7      3A     9D     FF      LDA TEMP         ;NO - RECALL CHAR TO BE ADDE
7777      1CDA      12     .      .      STAX D           ;PUT CHARACTER IN NEW BLOCK
7778      1CDB      .      .      .      ;              (OVERWRITE EOL)
7779      1CDB      C9     .      .      RET              ;RETURN
7780      1CDC      .      .      .      ;*****
7781      1CDC      .      .      .      ; BLOCK NOT AVAILABLE *
7782      1CDC      .      .      .      ; WRITE EOL AT END OF LAST BLOCK *
7783      1CDC      .      .      .      ;*****
7784      1CDC      .      .      .      CRI260 EQU $
7785      1CDC      2A     94     FF      LHLD EOLADR      ;GET ADR OF CHR BEFORE LNK
7786      1CDF      36     CC     .      MVI M,EOL        ;WRITE EOL
7787      1CE1      C9     .      .      RET              ;RETURN
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 241
7789	1CE2	. . .	;*****	
7790	1CE2	. . .	; RIGHT MARGIN OR END OF LINE REACHED - *	
7791	1CE2	. . .	; TERMINATE AND OPTIONALLY PUSH CHARACTERS *	
7792	1CE2	. . .	; TO THE NEXT LINE (WRAP AROUND) *	
7793	1CE2	. . .	;*****	
7794	1CE2	. . .	CRI300 EQU \$	
7795	1CE2	3A 89 FF	LDA DCHAR ;GET THE INSERTED CHARACTER	
7796	1CE5	B7 . .	ORA A ;IS IT ASCII?	
7797	1CE6	FA 4A 1C	JM CRI140 ;NO - CONTINUE INSERTING	
7798	1CE9	79 . .	MOV A,C ;YES - RECALL ENDING COLUMN	
7799	1CEA	. . .	CRI305 EQU \$	
7800	1CEA	32 D7 FF	STA PARM5 ;SAVE ENDING COLUMN NUMBER	
7801	1CED	22 96 FF	SHLD LNKSAV ;SAVE ENDING CHARACTER ADDR	
7802	1CF0	EB . .	XCHG ;PUT ENDING ADDRESS IN D,E	
7803	1CF1	CD 97 7D	CALL INITD1 ;INIT CHAR BUFFER POINTERS	
7804	1CF4	13 . .	INX D ;GET ADDRESS OF NEXT EXCESS	
7805	1CF5	CD 90 0C	CALL NXTCHR ;CHARACTER	
7806	1CF8	EB . .	XCHG	
7807	1CF9	22 D5 FF	SHLD PARM6 ;ARE WE AT AN EOL LINK?	
7808	1CFC	78 . .	MOV A,B ;(PUT 1ST EXCESS CHAR IN A	
7809	1CFD	CA 1C 1D	JZ CRI320 ;NO - ACCUMULATE EXCESS	
7810	1D00	CD 04 15	CALL A2OUTB ;YES - SAVE FIRST EXCESS CHA	
7811	1D03	B7 . .	ORA A ;IS IT ASCII?	
7812	1D04	F2 31 1D	JP CRI330 ;YES - CHECK FOR INSERT WRAP	
7813	1D07	C9 . .	RET ;NO - RETURN	
7814	1D08	. . .	;	
7815	1D08	. . .	; ACCUMULATE THE EXCESS CHARACTERS	
7816	1D08	. . .	;	
7817	1D08	. . .	CRI310 EQU \$	
7818	1D08	2A D5 FF	LHLD PARM6 ;RECALL EXCESS CHAR ADDRESS	
7819	1D08	EB . .	XCHG ;PUT ADDRESS INTO D,E	
7820	1D0C	3E FF .	MVI A,-1 ;SET DELETED CHAR TO -1	
7821	1D0E	32 98 FF	STA CHSAV	
7822	1D11	0E 50 .	MVI C,MAXCOL+1 ;FORCE DELETE PAST MARGIN	
7823	1D13	CD 72 1B	CALL CHRDL1 ;DELETE ONE EXCESS CHARACTER	
7824	1D16	3A 98 FF	LDA CHSAV ;RECALL THE DELETED CHARACTE	
7825	1D19	47 . .	MOV B,A ;SAVE THE CHARACTER IN B-REG	
7826	1D1A	04 . .	INR B ;ANY CHARACTER DELETED?	
7827	1D1B	C8 . .	RZ ;NO - RETURN (A#0)	
7828	1D1C	. . .	CRI320 EQU \$;YES - ACCUMULATE EXCESS	
7829	1D1C	CD 04 15	CALL A2OUTB ;PUT DELETED CHAR INTO BUFFE	
7830	1D1F	B7 . .	GRA A ;WAS DELETED CHARACTER ASCII	
7831	1D20	FA 08 1D	JM CRI310 ;NO - CONTINUE ACCUMULATING	
7832	1D23	3A D7 FF	LDA PARM5 ;RECALL ENDING COLUMN NUMBER	
7833	1D26	FE 50 .	CPI MAXCOL+1 ;TERMINATE ON LAST COLUMN?	
7834	1D28	2A 96 FF	LHLD LNKSAV ;(RECALL ENDING CHAR ADDR)	
7835	1D2B	EB . .	XCHG	
7836	1D2C	3E C3 .	MVI A,FILL ;(SET FOR FILL PAD)	
7837	1D2E	CC B7 1D	CZ CLERLO ;YES - CLEAR REST OF LINE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 242
=====
7839     1D31     . . .      ;
7840     1D31     . . .      ; EXCESS CHARACTERS ACCUMULATED - CHECK FOR WRAP
7841     1D31     . . .      ;
7842     1D31     . . .      CRI330 EQU $
7843     1D31     3A F8 FF    LDA CMFLGS      ;GET THE COMMON FLAGS
7844     1D34     2F . .     CMA             ;COMPLEMENT FLAGS
7845     1D35     E6 02 .     ANI INSWRP     ;WRAP AROUND ENABLED?
7846     1D37     C0 . .     RNZ            ;NO - RETURN (A#0)
7847     1D38     3A C1 FF    LDA CURCOL     ;YES - GET THE CURRENT COLUMN
7848     1D3B     47 . .     MOV B,A        ;SAVE VALUE IN B-REGISTER
7849     1D3C     3A BE FF    LDA RHTMGN
7850     1D3F     B8 . .     CMP B          ;CURSOR BEYOND RIGHT MARGIN?
7851     1D40     D8 . .     RC             ;YES - RETURN
7852     1D41     CD CF 1A   CALL CHKFMS    ;FORMAT/SOFT KEY DEFINE MODE
7853     1D44     C0 . .     RNZ            ;YES - RETURN
7854     1D45     C5 . .     PUSH B        ;NO - SAVE CURRENT COLUMN AN
7855     1D46     21 C0 FF    LXI H,CURROW  ;INCREMENT TO NEXT ROW
7856     1D49     34 . .     INR M
7857     1D4A     2A C9 FF    LHLD LSTLIN   ;CHECK TO SEE IF NEXT LINE
7858     1D4D     5E . .     MOV E,M       ;IS FULL (I.E., NO "EOL"
7859     1D4E     23 . .     INX H         ;BEFORE RIGHT MARGIN)
7860     1D4F     56 . .     MOV D,M
7861     1D50     1C . .     INR E         ;DOES NEXT LINE EXIST?
7862     1D51     1D . .     DCR E         ;(LSB # 0)?
7863     1D52     CA 65 1D   JZ CRI400     ;NO - ADD CHAR TO NEW LINE
7864     1D55     13 . .     INX D         ;YES - START FROM BEGINNING
7865     1D56     3A BE FF    LDA RHTMGN    ;OF LINE TO RIGHT MARGIN
7866     1D59     CD D6 20   CALL FNDLS0   ;NEXT LINE FULL?
7867     1D5C     F2 65 1D   JP CRI400     ;NO - ADD OVERFLOW CHARACTER
7868     1D5F     . . .      ; TO NEXT LINE
7869     1D5F     CD 00 0B   CALL LININS   ;YES - INSERT A LINE
7870     1D62     CA 7E 1D   JZ CRI450     ;EXIT IF MEMORY LOCKED
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 243
7872	1D65	.	.	;	
7873	1D65	.	.	; INSERT CHARACTERS INTO NEXT LINE	
7874	1D65	.	.	;	
7875	1D65	.	.	CRI400 EQU \$	
7876	1D65	21	3B	FF LXI H,B2DEND ;GET BUFFER POINTER	
7877	1D68	7E	.	MOV A,M	
7878	1D69	FE	3C	. CPI B2DBFL-1 ;ALL BYTES DONE?	
7879	1D6B	CA	7E	1D JZ CRI450 ;YES - EXIT	
7880	1D6E	35	.	. DCR M ;NO - UPDATE BUFFER POINTER	
7881	1D6F	6F	.	. MOV L,A ;PUT LSB INTO L	
7882	1D70	3A	BF	FF LDA LFTMGN ;SET TO INSERT CHARACTER AT	
7883	1D73	32	C1	FF STA CURCOL ;LEFT MARGIN	
7884	1D76	7E	.	. MOV A,M ;GET CHARACTER TO INSERT	
7885	1D77	CD	1A	1C CALL CHRINS ;INSERT CHARACTER	
7886	1D7A	B7	.	. ORA A ;INSERT SUCCESSFUL?	
7887	1D7B	C2	65	1D JNZ CRI400 ;YES - DO NEXT BYTE	
7888	1D7E	.	.	. ;	
7889	1D7E	.	.	. ; ALL CHARACTERS INSERTED - EXIT	
7890	1D7E	.	.	. ;	
7891	1D7E	.	.	. CRI450 EQU \$	
7892	1D7E	21	C0	FF LXI H,CURROW	
7893	1D81	35	.	. DCR M ;RESTORE THE ROW NUMBER	
7894	1D82	F1	.	. POP PSW ;RECALL THE COLUMN NUMBER	
7895	1D83	23	.	. INX H	
7896	1D84	77	.	. MOV M,A ;RESTORE COLUMN NUMBER	
7897	1D85	3C	.	. INR A ;FORCE A # 0	
7898	1D86	C9	.	. RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 244
=====
7900      1D87      . . .      ;*****
7901      1D87      . . .      ; CRI500 - GET PREVIOUS CHARACTER *
7902      1D87      . . .      ;*****
7903      1D87      . . .      ;
7904      1D87      . . .      ; ENTRY:  H,L = CURRENT CHARACTER ADDRESS
7905      1D87      . . .      ;          LNKSAV = ADDRESS OF MSB PART OF NEXT
7906      1D87      . . .      ;          BLOCK LINK IN PREVIOUS BLOCK
7907      1D87      . . .      ;
7908      1D87      . . .      ; EXIT :  A = PREVIOUS CHARACTER
7909      1D87      . . .      ;          H,L = ADDRESS OF PREVIOUS CHARACTER
7910      1D87      . . .      ;          P = CHARACTER IS ASCII
7911      1D87      . . .      ;          M = CHARACTER IS NON-DISPLAY CONTROL
7912      1D87      . . .      ;
7913      1D87      . . .      CRI500 EQU $
7914      1D87      23 . . .      INX H          ;MOVE TO PREVIOUS CHARACTER
7915      1D88      7D . . .      MOV A,L        ;IN BLOCK
7916      1D89      E6 0F . . .      ANI BLKSM      ;PREVIOUS CHARACTER IN BLOCK
7917      1D8B      C2 92 1D . . .      JNZ CRI510     ;YES - GET IT
7918      1D8E      2A 96 FF . . .      LHL LNKSAV     ;NO - GET PREV BLOCK ADDRESS
7919      1D91      23 . . .      INX H          ;SET TO LAST CHARACTER ADDR
7920      1D92      . . .      CRI510 EQU $
7921      1D92      7E . . .      MOV A,M        ;GET THE PREVIOUS CHARACTER
7922      1D93      B7 . . .      ORA A          ;SET FLAGS FOR ASCII OR NOT
7923      1D94      C9 . . .      RET            ;ASCII AND RETURN
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
7925	1D95	.	.	*****
7926	1D95	.	.	; CLEARL - CLEAR LINE *
7927	1D95	.	.	*****
7928	1D95	.	.	;
7929	1D95	.	.	; ENTRY: DON'T CARE
7930	1D95	.	.	;
7931	1D95	.	.	; EXIT : A = -1, ROW NOT FOUND
7932	1D95	.	.	; = 0, CHARACTER FOUND AND CLEAR DONE
7933	1D95	.	.	; > 0, COLUMN PAST EOL, CLEAR NOT DONE
7934	1D95	.	.	;
7935	1D95	.	.	CLEARL EQU \$
7936	1D95	CD	CA 07	CALL RCADR4 ;DOES ROW EXIST?
7937	1D98	C0	.	RNZ ;NO - RETURN
7938	1D99	CD	CF 1A	CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE
7939	1D9C	CA	AD 1D	JZ CLERLA ;NO - DO NORMAL CLEAR LINE
7940	1D9F	F2	F3 1D	JP CLL400 ;FORMAT MODE - CLEAR FIELD
7941	1DA2	.	.	*****
7942	1DA2	.	.	; SOFT KEY DEFINE MODE - CLEAR DATA ROWS ONLY *
7943	1DA2	.	.	*****
7944	1DA2	3A	C0 FF	LDA CURROW ;GET CURSOR ROW
7945	1DA5	0F	.	RRC ;IN DATA LINE (ODD ROW #)?
7946	1DA6	D0	.	RNC ;NO - INHIBIT CLEAR
7947	1DA7	1A	.	LDAX D ;GET FIRST CHARACTER
7948	1DA8	FE	C1 .	CPI ENDPR ;END PROTECT?
7949	1DAA	CC	90 0C	CZ NXTCHR ;YES - SKIP TO 1ST ASCII CHA
7950	1DAD	.	.	CLERLA EQU \$
7951	1DAD	CD	E5 1A	CALL CHKSFK ;SOFT KEY DEFINE MODE?
7952	1DB0	3E	0C .	MVI A,SETFRN ;(SET CONTROL CODE)
7953	1DB2	CC	08 48	CZ ZKBCTL ;NO - UPDATE FOREIGN MODE
7954	1DB5	3E	CC .	MVI A,EOL ;CLEAR LINE WITH "EOL" ENDIN

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 246
=====
7956     1DB7      . . .      ;*****
7957     1DB7      . . .      ; CLERL0 - CLEAR REST OF LINE *
7958     1DB7      . . .      ;*****
7959     1DB7      . . .      ;
7960     1DB7      . . .      ; ENTRY:  A = TERMINATOR CHARACTER
7961     1DB7      . . .      ;          D,E = CLEAR STARTING ADDRESS
7962     1DB7      . . .      ;
7963     1DB7      . . .      ; EXIT :  SEE "CLEARL"
7964     1DB7      . . .      ;
7965     1DB7      . . .      CLERL0 EQU $
7966     1DB7      2A C9 FF    LHLD LSTLIN    ;GET CURRENT LINE ADDRESS
7967     1DBA      . . .      CLERL1 EQU $
7968     1DBA      32 8F FF    STA FILCHR     ;SAVE TERMINATOR CHARACTER
7969     1DBD      44 . .     MOV B,H        ;SET B,C TO ADDRESS OF NEXT
7970     1DBE      4D . .     MOV C,L        ;LINE POINTER'S LSB
7971     1DBF      7B . .     MOV A,E        ;SET H,L TO ADDRESS OF NEXT
7972     1DC0      E6 F0 . ANI 377Q-BLKSM ;BLOCK LINK IN CURRENT
7973     1DC2      6F . .     MOV L,A        ;BLOCK
7974     1DC3      62 . .     MOV H,D
7975     1DC4      7E . .     MOV A,M        ;GET NEXT BLOCK
7976     1DC5      03 . .     INX B          ;SET B,C TO MSB OF NEXT LINE
7977     1DC6      71 . .     MOV M,C        ;POINTER
7978     1DC7      23 . .     INX H
7979     1DC8      4E . .     MOV C,M
7980     1DC9      70 . .     MOV M,B
7981     1DCA      45 . .     MOV B,L        ;SAVE LSB OF LINK'S MSB ADDR
7982     1DCB      61 . .     MOV H,C
7983     1DCC      6F . .     MOV L,A
7984     1DCD      E5 . .     PUSH H        ;SAVE ADDRESS OF NEXT BLOCK
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 247
7986	1DCE	.	.	*****	
7987	1DCE	.	.	; INSERT FILL CHARS BETWEEN LINK AND EOL *	
7988	1DCE	.	.	*****	
7989	1DCE	7B	.	MOV A,E ;COMPUTE NO. OF FILLS	
7990	1DCF	E6	0F	ANI BLKSM	
7991	1DD1	06	02	SUI 2 ;LESS THAN 2?	
7992	1DD3	FA	E3 1D	JM CLL160 ;YES - RELEASE THE BLOCK	
7993	1DD6	58	.	MOV E,B ;SET H,L TO ADDRESS OF MSB	
7994	1DD7	EB	.	XCHG ;PART OF NEXT BLOCK POINTE	
7995	1DD8	.	.	CLL120 EQU \$	
7996	1DD8	23	.	INX H ;ADVANCE TO NEXT BYTE	
7997	1DD9	36	C3	MVI M,FILL ;STORE FILL CHARACTER	
7998	1DD8	3D	.	DCR A ;ALL BYTES DONE?	
7999	1DDC	F2	08 1D	JP CLL120 ;NO - CONTINUE FILLING	
8000	1DDF	3A	8F FF	LDA FILCHR ;YES - GET AND STORE FINAL	
8001	1DE2	77	.	MOV M,A ;FILL CHARACTER	
8002	1DE3	.	.	*****	
8003	1DE3	.	.	; RELEASE EXCESS DISPLAY BLOCKS *	
8004	1DE3	.	.	*****	
8005	1DE3	.	.	CLL160 EQU \$	
8006	1DE3	D1	.	POP D ;RECALL ADDRESS OF NEXT BLOC	
8007	1DE4	7B	.	MOV A,E	
8008	1DE5	2F	.	CMA ;IS THE LINK AN EOL LINK	
8009	1DE6	E6	0F	ANI BLKSM ;(LOW 4 BITS NOT ALL ONES)	
8010	1DE8	C2	F1 1D	JNZ CLL310 ;YES - EXIT	
8011	1DEB	1B	.	DCX D ;NO - ADD BLOCKS TO FREE LIS	
8012	1DEC	1B	.	DCX D ;SET ADDRESS TO LSB OF NEXT	
8013	1DED	1B	.	DCX D ;LINE FIELD IN FIRST BLOCK	
8014	1DEE	CD	8E 07	CALL PUTLIN ;ADD BLOCKS TO FREE LIST	
8015	1DF1	.	.	CLL310 EQU \$	
8016	1DF1	AF	.	XRA A ;SET ZERO FLAG FOR CLEARS	
8017	1DF2	C9	.	RET ;RETURN	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 248
8019	1DF3	. . .	;*****	
8020	1DF3	. . .	; CLEAR LINE FUNCTION FOR FORMAT MODE *	
8021	1DF3	. . .	;*****	
8022	1DF3	. . .	CLL400 EQU \$	
8023	1DF3	04 . .	INR B ;CURSOR IN PROTECTED FIELD?	
8024	1DF4	C8 . .	RZ ;YES - RETURN, DON'T DO CLEA	
8025	1DF5	. . .	;*****	
8026	1DF5	. . .	; CLEAR UNPROTECTED FIELD *	
8027	1DF5	. . .	; D,E = ADDRESS OF FIRST ASCII CHAR IN FIELD *	
8028	1DF5	. . .	;*****	
8029	1DF5	. . .	CLER01 EQU \$	
8030	1DF5	EB . .	XCHG	
8031	1DF6	. . .	CLER02 EQU \$	
8032	1DF6	EB . .	XCHG ;PUT CHARACTER ADDR INTO D,E	
8033	1DF7	13 . .	INX D ;SET TO PREVIOUS CHARACTER	
8034	1DF8	. . .	CLL510 EQU \$	
8035	1DF8	CD 90 0C	CALL NXTCHR ;GET THE NEXT CHARACTER	
8036	1DFB	C2 23 1E	JNZ CLL580 ;CHECK EXIT IF EOL LINK	
8037	1DFE	87 . .	ADD A ;ASCII?	
8038	1DFF	DA 09 1E	JC CLL540 ;NO - CONTINUE	
8039	1E02	3E 20 .	MVI A,ABLNK ;YES - STORE BLANK	
8040	1E04	12 . .	STAX D	
8041	1E05	0C . .	INR C ;INCREMENT COLUMN	
8042	1E06	C3 F8 1D	JMP CLL510 ;TRY NEXT CHARACTER	
8043	1E09	. . .	;*****	
8044	1E09	. . .	; NON-ASCII CHARACTER *	
8045	1E09	. . .	;*****	
8046	1E09	. . .	CLL540 EQU \$	
8047	1E09	FA 12 1E	JM CLL550 ;NOT DSPLY CNTRL - CHECK MORE	
8048	1E0C	. . .	;*****	
8049	1E0C	. . .	; DELETE DISPLAY ENHANCEMENT CHAR *	
8050	1E0C	. . .	;*****	
8051	1E0C	. . .	CLL544 EQU \$	
8052	1E0C	CD 11 1C	CALL CHRDL2 ;DELETE ENHANCEMENT CODE	
8053	1E0F	C3 F8 1D	JMP CLL510 ;CONTINUE CLEARING	

				=====		PAGE 249	
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS		
=====							
8055	1E12	.	.	.	*****		
8056	1E12	.	.	.	; NOT ASCII OR DISPLAY CONTROL *		
8057	1E12	.	.	.	*****		
8058	1E12	.	.	.	CLL550 EQU \$		
8059	1E12	1F	.	.	RAR	;RESTORE CHARACTER	
8060	1E13	FE	C3	.	CPI FILL	;END OF LINE FILL?	
8061	1E15	CA	F8	1D	JZ CLL510	;YES - GO TO NEXT CHARACTER	
8062	1E18	FE	C0	.	CPI STPR	;START PROTECT?	
8063	1E1A	C8	.	.	RZ	;YES - TERMINATE CLEAR	
8064	1E1B	FE	C5	.	CPI STPFLG+1	;FORMAT CONTROL CODE?	
8065	1E1D	DA	0C	1E	JC CLL544	;YES - DELETE IT	
8066	1E20	C3	F8	1D	JMP CLL510	;NO - GO TO NEXT CHARACTER	
8067	1E23	.	.	.	*****		
8068	1E23	.	.	.	; LINK FOUND *		
8069	1E23	.	.	.	; MOVE TO NEXT BLOCK *		
8070	1E23	.	.	.	*****		
8071	1E23	.	.	.	CLL580 EQU \$		
8072	1E23	1A	.	.	LDAX D	;GET NEXT LINE LINK'S MSB	
8073	1E24	FE	CE	.	CPI EOP	;END OF DISPLAY LIST?	
8074	1E26	C8	.	.	RZ	;YES - RETURN	
8075	1E27	CD	FA	1F	CALL FLDSR2	;CONTINUATION FIELD?	
8076	1E2A	CA	F8	1D	JZ CLL510	;YES - CONTINUE CLEAR	
8077	1E2D	AF	.	.	XRA A	;NO - TERMINATE CLEAR AND	
8078	1E2E	C9	.	.	RET	;RETURN END ON END OF FIEL	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 250
=====
8080      1E2F      . . .      ;*****
8081      1E2F      . . .      ; DSPMSG - DISPLAY MESSAGE *
8082      1E2F      . . .      ;*****
8083      1E2F      . . .      ;
8084      1E2F      . . .      ; ENTRY:  NC - ADD MESSAGE TO NORMAL DISPLAY
8085      1E2F      . . .      ;          C - REPLACE DISPLAY WITH MESSAGE
8086      1E2F      . . .      ;          MSGPT1-MSGPT8 = POINTERS TO MESSAGE
8087      1E2F      . . .      ;          SECTIONS
8088      1E2F      . . .      ;
8089      1E2F      . . .      ; EXIT :  ALL REGISTERS DESTROYED
8090      1E2F      . . .      ;
8091      1E2F      . . .      DSPMSG EQU $          ;SET C-FLAG TO FORCE DISPLAY
8092      1E2F      37 . .      STC                  ;REPLACEMENT BY MESSAGE
8093      1E30      . . .      DSPMSG1 EQU $
8094      1E30      22 F1 FF      SHLD MSGPT1         ;SET MESSAGE POINTER 1
8095      1E33      . . .      DSPMSG EQU $
8096      1E33      D2 57 1E      JNC DSM500          ;ADD MESSAGE TO DISPLAY
8097      1E36      . . .      ;***** GRAPHICS MODIFICATION *****
8098      1E36      CD 50 60      CALL ZVID1          ;GRAPHICS OFF, ALLOW A/N
8099      1E39      . . .      ;*****
8100      1E39      . . .      DSPMSG2 EQU $
8101      1E39      01 4F FE      LXI B,DSPSTR        ;SET DESTINATION POINTER
8102      1E3C      21 F2 FF      LXI H,MSGPT1+1     ;SET INITIAL TABLE POINTER
8103      1E3F      . . .      ;
8104      1E3F      . . .      ; TRANSFER MESSAGE TO MESSAGE BUFFER
8105      1E3F      . . .      ;
8106      1E3F      . . .      DSM010 EQU $
8107      1E3F      56 . .      MOV D,M             ;GET POINTER TO MESSAGE
8108      1E40      2B . .      DCX H
8109      1E41      5E . .      MOV E,M
8110      1E42      2B . .      DCX H              ;SET TO NEXT POINTER
8111      1E43      EB . .      XCHG               ;PUT POINTER INTO H,L
8112      1E44      CD 29 0C      CALL MOVCHR         ;XFR MESSAGE PART TO BUFFER
8113      1E47      EB . .      XCHG               ;PUT POINTER TO TABLE IN H,L
8114      1E48      CA 3F 1E      JZ DSM010          ;DO NEXT PART IF NOT EOP END
8115      1E4B      21 4F FE      LXI H,DSPSTR        ;SET DISPLAY POINTER TO
8116      1E4E      22 FE FF      SHLD DISPST         ;MESSAGE AREA
8117      1E51      3E 18 .      MVI A,MAXROW+1     ;REMOVE CURSOR FROM DISPLA
8118      1E53      32 20 87      STA IOCRWR
8119      1E56      C9 . .      RET                ;RETURN
=====

```


13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 252
=====
8171     1E84     2A  CB  FF      LHLD TOPLIN      ;GET TOP LINE ADDRESS
8172     1E87     2B  .   .      DCX  H           ;SET TO FIRST CHAR ADDRESS
8173     1E88     22  FE  FF      SHLD DISPST     ;SET DISPLAY START POINTER
8174     1E8B     C3  D3  10     JMP  DISLN1     ;SET THE DISPLAY CURSOR
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 253
8176	1E8E	.	.	*****	
8177	1E8E	.	.	; FORMON - ENTER FORMAT MODE *	
8178	1E8E	.	.	*****	
8179	1E8E	.	.	FORMON EQU \$	
8180	1E8E	CD	78 11	CALL CKREDIT ;EDIT MODE?	
8181	1E91	C0	.	RNZ ;YES - INHIBIT FORMAT MODE	
8182	1E92	21	4F 00	LXI H,MAXCOL ;NO - SET MARGINS TO ENDS OF	
8183	1E95	22	8E FF	SHLD RHTMGN ;DISPLAY	
8184	1E98	3E	08 .	MVI A,FORMAT ;TURN ON FORMAT MODE FLAG	
8185	1E9A	CD	0E 48	CALL ZSTMD1	
8186	1E9D	.	.	;	SET CURSOR TO FIRST
8187	1E9D	.	.	;	UNPROTECTED FIELD
8188	1E9D	.	.	;	
8189	1E9D	.	.	*****	
8190	1E9D	.	.	; CURPH - CURSOR POINTER HOME (UP) *	
8191	1E9D	.	.	*****	
8192	1E9D	.	.	CURPH EQU \$	
8193	1E9D	3E	FE .	MVI A,377Q-SDACOM ;CLEAR DATACOM INPUT	
8194	1E9F	CD	53 17	CALL CLRDFL ;FLAG TO DISABLE TRANSMIT-	ONLY FIELDS
8195	1EA2	.	.	;	
8196	1EA2	.	.	;	
8197	1EA2	.	.	CURPH1 EQU \$	
8198	1EA2	CD	7C 23	CALL CURPRT ;SET CURSOR TO LEFT MARGIN	
8199	1EA5	CD	E5 1A	CALL CHKSEK ;SOFT KEY MODE?	
8200	1EA8	C2	08 1F	JNZ HUP060 ;YES - SET CURSOR ONLY	
8201	1EAB	32	A3 FF	STA TLINO ;NO - SET TOP LINE # TO ZERO	
8202	1EAE	3D	.	DCR A ;RESET SPOW LATCH	
8203	1EAF	32	6C FF	STA SPOWL	
8204	1EB2	CD	F7 0B	CALL MLKSCH ;DISPLAY AREA LOCKED?	
8205	1EB5	CA	19 1F	JZ HUP100 ;NO - HOME TO FIRST LINE	
8206	1EB8	.	.	;	
8207	1EB8	.	.	;	DISPLAY LOCK ON - CHANGE ONLY UNLOCKED LINES
8208	1EB8	.	.	;	
8209	1EB8	54	.	MOV D,H ;SAVE ADDRESS OF LSB PART OF	
8210	1EB9	5D	.	MOV E,L ;NEXT LINE POINTER IN FIRS	
8211	1EBA	23	.	INX H ;UNLOCKED LINE	
8212	1EBB	23	.	INX H ;GET ADDRESS OF LAST LOCKED	
8213	1EBC	4E	.	MOV C,M ;ROW	
8214	1EBD	23	.	INX H	
8215	1EBE	46	.	MOV B,M	
8216	1EBF	2A	CB FF	LHLD TOPLIN ;GET PTR TO TOP DSPLY LINE	
8217	1EC2	23	.	INX H ;GET ADDRESS OF FIRST LINE	
8218	1EC3	23	.	INX H ;ABOVE TOP DISPLAY LINE	
8219	1EC4	7E	.	MOV A,M	
8220	1EC5	87	.	ORA A ;ANY LINES ABOVE DISPLAY?	
8221	1EC6	CA	F3 1E	JZ HUP050 ;NO - POSITION CURSOR ONLY	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 254
8223	1EC9	.	.	.	;
8224	1EC9	.	.	.	;
8225	1EC9	.	.	.	;
8226	1EC9	36	00	.	MVI M,0 ;ZERO PREV LINE PTR OF TOP L
8227	1ECB	23	.	.	INX H
8228	1ECC	66	.	.	MOV H,M ;SET H,L TO FIRST LINE ABOVE
8229	1ECD	6F	.	.	MOV L,A ;DISPLAY
8230	1ECE	1B	.	.	DCX D ;SET ITS NEXT LINE POINTER T
8231	1ECF	73	.	.	MOV M,E ;FIRST CHARACTER OF FIRST
8232	1ED0	23	.	.	INX H ;UNLOCKED LINE
8233	1ED1	72	.	.	MOV M,D
8234	1ED2	EB	.	.	XCHG ;SET PREVIOUS LINE POINTER O
8235	1ED3	23	.	.	INX H ;FIRST UNLOCKED LINE TO
8236	1ED4	23	.	.	INX H ;FIRST LINE ABOVE DISPLAY
8237	1ED5	23	.	.	INX H
8238	1ED6	77	.	.	MOV M,A
8239	1ED7	23	.	.	INX H
8240	1ED8	72	.	.	MOV M,D
8241	1ED9	2A	9F	FF	LHLD FLINE ;REPLACE CONTENTS OF FLINE
8242	1EDC	EB	.	.	XCHG ;WITH VALUES FROM TOPLIN
8243	1EDD	2A	CB	FF	LHLD TOPLIN
8244	1EE0	22	9F	FF	SHLD FLINE
8245	1EE3	62	.	.	MOV H,D ;SET PREVIOUS LINE POINTER O
8246	1EE4	6B	.	.	MOV L,E ;CURRENT TOP LINE TO POINT
8247	1EE5	23	.	.	INX H ;TO LAST LOCKED ROW
8248	1EE6	23	.	.	INX H
8249	1EE7	71	.	.	MOV M,C
8250	1EE8	23	.	.	INX H
8251	1EE9	70	.	.	MOV M,B
8252	1EEA	60	.	.	MOV H,B ;SET H,L TO MSB PART OF NEXT
8253	1EEB	69	.	.	MOV L,C ;LINE POINTER IN LAST
8254	1EEC	23	.	.	INX H ;LOCKED ROW
8255	1EED	42	.	.	MOV B,D ;SET NEXT LINE POINTER TO
8256	1EEE	4B	.	.	MOV C,E ;POINT TO FIRST CHARACTER
8257	1EEF	0B	.	.	DCX B ;OF LINE POINTED BY FLINE
8258	1EF0	CD	CA	10	CALL DISLNK
8259	1EF3	.	.	.	;
8260	1EF3	.	.	.	;
8261	1EF3	.	.	.	;
8262	1EF3	.	.	.	HUP050 EQU \$
8263	1EF3	CD	5F	0D	CALL ROLUP3 ;SET "LSTLIN" AND "CURADR"
8264	1EF6	CD	D4	1A	CALL CHKFMT ;FORAMT MODE?
8265	1EF9	EE	08	.	XRI FORMAT ;(REVERSE RESULT OF TEST)
8266	1EFB	CA	1C	1F	JZ HUP110 ;YES - LOCATE FIRST FIELD
8267	1EFE	.	.	.	;
8268	1EFE	.	.	.	;
8269	1EFE	3A	6B	FF	LDA MLKROW ;NO - SET CURSOR TO FIRST
8270	1F01	32	C0	FF	STA CURROW ;UNLOCKED ROW
8271	1F04	32	C7	FF	STA LSTROW
8272	1F07	C9	.	.	RET ;RETURN

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
8274	1F08	.	.	;
8275	1F08	.	.	; DEFINE SOFT KEYS HOME UP
8276	1F08	.	.	;
8277	1F08	.	.	HUP060 EQU \$
8278	1F08	AF	.	XRA A ;SET CURSOR ROW TO ZERO
8279	1F09	32	C0	FF STA CURROW
8280	1F0C	32	C7	FF STA LSTROW
8281	1F0F	2A	A6	FF LHLD SFTKYS ;SET "CURADR" AND "LSTLIN"
8282	1F12	23	.	INX H ;TO FIRST SOFT KEY LINE
8283	1F13	CD	60	0D CALL ROLUPC
8284	1F16	C3	2F	1F JMP FLDSR1 ;LOCATE FIRST FIELD
8285	1F19	.	.	;
8286	1F19	.	.	; DISPLAY NOT LOCKED - SET TOPLIN TO FLINE
8287	1F19	.	.	;
8288	1F19	.	.	HUP100 EQU \$
8289	1F19	3A	6B	FF LDA MLKROW ;SET CURSOR TO 1ST UNLK RW
8290	1F1C	.	.	HUP110 EQU \$
8291	1F1C	32	C0	FF STA CURROW ;SET NEW CURRENT ROW
8292	1F1F	AF	.	XRA A
8293	1F20	32	C7	FF STA LSTROW ;SET LAST ROW DONE TO ZERO
8294	1F23	57	.	MOV D,A ;SET D=0 TO FLAG TLINO UPDAT
8295	1F24	21	9F	FF LXI H,FLINE
8296	1F27	7E	.	MOV A,M ;SET TOP LINE POINTER TO
8297	1F28	CD	77	0D CALL ROLUP1 ;FIRST DISPLAY LINE
8298	1F2B	CD	CF	1A CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE
8299	1F2E	C8	.	RZ ;NO - RETURN
8300	1F2F	.	.	;
8301	1F2F	.	.	;
8302	1F2F	.	.	;

YES - FALL INTO "FLDSR1" TO
FIND FIRST UNPROTECTED
FIELD

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 256
8304	1F2F	.	.	. ;*****	
8305	1F2F	.	.	. ; FLDSR - LOCATE THE NEXT UNPROTECTED FIELD *	
8306	1F2F	.	.	. ;*****	
8307	1F2F	.	.	. ;	
8308	1F2F	.	.	. ; ENTRY: DON'T CARE	
8309	1F2F	.	.	. ;	
8310	1F2F	.	.	. ; EXIT : NZ - FIELD FOUND	
8311	1F2F	.	.	. ; D,E = ADDRESS OF "ENDPR"	
8312	1F2F	.	.	. ; CURADR,CURCOL,CURROW,LSTLIN,LSTCOL	
8313	1F2F	.	.	. ; LSTROW UPDATE TO CORRESPOND TO	
8314	1F2F	.	.	. ; FIELD FOUND	
8315	1F2F	.	.	. ; Z - FIELD NOT FOUND	
8316	1F2F	.	.	. ; ALL REGISTERS DESTROYED	
8317	1F2F	.	.	. ;	
8318	1F2F	.	.	. FLDSR1 EQU \$;LOOK FOR NEXT UNPROTECT	
8319	1F2F	21	DA	FF LXI H,NEWROW ;INITIALIZE ROW COUNT	
8320	1F32	36	00	. MVI M,0 ;TO ZERO	
8321	1F34	2E	C1	. MVI L,CURCOL-BASE ;GET CURRENT COLUMN	
8322	1F36	4E	.	. MOV C,M ;POSITION	
8323	1F37	C3	58	1F JMP FSR100	
8324	1F3A	.	.	. FLDSR EQU \$	
8325	1F3A	AF	.	. XRA A ;ZERO NUMBER OF ROWS ROLLED	
8326	1F3B	32	DA	FF STA NEWROW	
8327	1F3E	CD	A9	07 CALL RCADRB ;DOES CURSOR ROW EXIST?	
8328	1F41	FA	0B	08 JM ZRETRN ;NO - RETURN ZERO	
8329	1F44	4F	.	. MOV C,A ;YES - SAVE LAST COLUMN FOUR	
8330	1F45	CD	90	11 CALL CKPROT ;CURSOR IN PROTECTED FIELD?	
8331	1F48	CA	58	1F JZ FSR100 ;YES - LOOK FOR NEXT UNPROTC	
8332	1F4B	.	.	. ;*****	
8333	1F4B	.	.	. ; CURSOR IS IN UNPROTECTED FIELD *	
8334	1F4B	.	.	. ; SEARCH FOR START OF NEXT PROTECTED FIELD *	
8335	1F4B	.	.	. ;*****	
8336	1F4B	.	.	. FSR080 EQU \$	
8337	1F4B	21	C0	C0 LXI H,STPR*256+STPR	
8338	1F4E	CD	42	20 CALL FNDCU1 ;ANY MORE FIELDS IN LINE?	
8339	1F51	CA	5E	1F JZ FSR120 ;NO - GO TO NEXT LINE	
8340	1F54	.	.	. ;*****	
8341	1F54	.	.	. ; ADVANCE CURSOR TO START OF PROTECTED FIELD *	
8342	1F54	.	.	. ;*****	
8343	1F54	3E	50	. MVI A,MAXCOL+1 ;COMPUTE NEW COLUMN	
8344	1F56	91	.	. SUB C	
8345	1F57	4F	.	. MOV C,A ;SAVE COLUMN IN C	
8346	1F58	.	.	. ;*****	
8347	1F58	.	.	. ; CURSOR IS IN PROTECTED FIELD *	
8348	1F58	.	.	. ; SEARCH FOR NEXT UNPROTECTED FIELD *	
8349	1F58	.	.	. ; IN THIS LINE *	
8350	1F58	.	.	. ;*****	
8351	1F58	.	.	. FSR100 EQU \$	
8352	1F58	CD	37	20 CALL FNDCU1 ;ANY MORE FIELDS IN LINE?	
8353	1F5B	C2	9C	1F JNZ FSR200 ;YES - SET CURSOR AND DISPLA	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 257
8355	1F5E	.	.	*****	
8356	1F5E	.	.	; NO MORE FIELDS IN LINE *	
8357	1F5E	.	.	; MOVE TO NEXT LINE *	
8358	1F5E	.	.	*****	
8359	1F5E	.	.	FSR120 EQU \$	
8360	1F5E	FE	C4	CPI STPFLG ;NON-DISPLAYING TERMINATOR?	
8361	1F60	CA	96 1F	JZ FSR140 ;YES - RETURN FAIL	
8362	1F63	4C	.	MOV C,H ;NO - SAVE TERMINATOR CHAR	
8363	1F64	CD	C1 1A	CALL CHAIN0 ;GET NEXT BLOCK LINK	
8364	1F67	7E	.	MOV A,M ;GET NEXT LINE LINK'S MSB	
8365	1F68	2B	.	DCX H	
8366	1F69	6E	.	MOV L,M ;PUT LSB INTO L-REGISTER	
8367	1F6A	FE	CE	CPI EOP ;END OF DISPLAY FOUND?	
8368	1F6C	CA	96 1F	JZ FSR140 ;YES - EXIT FIELD NOT FOUND	
8369	1F6F	67	.	MOV H,A ;NO - SAVE ADDRESS OF NEW	
8370	1F70	22	96 FF	SHLD LNKSAV ;LINE	
8371	1F73	EB	.	XCHG	
8372	1F74	21	DA FF	LXI H,NEWROW ;INCREMENT ROW NUMBER	
8373	1F77	34	.	INR M	
8374	1F78	AF	.	XRA A	
8375	1F79	32	C6 FF	STA LSTDCC ;CLEAR LAST DISPLAY CODE	
8376	1F7C	32	9D FF	STA TEMP	
8377	1F7F	79	.	MOV A,C ;GET LAST TERMINATOR CHAR	
8378	1F80	0E	00	MVI C,0 ;SET COLUMN TO ZERO	
8379	1F82	FE	C0	CPI STPR ;LOOKING FOR START PROTECT?	
8380	1F84	C2	58 1F	JNZ FSR100 ;NO - CONTINUE UNPROTECT FIN	
8381	1F87	.	.	YES - SEE IF CONTINUE UNPROT	
8382	1F87	.	.	*****	
8383	1F87	.	.	; SEARCH FOR PROTECTED FIELD *	
8384	1F87	.	.	; CHECK FOR CONTINUED UNPROTECTED FIELD *	
8385	1F87	.	.	*****	
8386	1F87	CD	FA 1F	CALL FLDSR2 ;FIRST CHAR AN "ENDPR"	
8387	1F8A	3A	9D FF	LDA TEMP ;(SET NEW LSTDCC VALUE)	
8388	1F8D	32	C6 FF	STA LSTDCC	
8389	1F90	CA	4B 1F	JZ FSR080 ;YES - LOOK FOR START PROTEC	
8390	1F93	C3	58 1F	JMP FSR100 ;NO - LOOK FOR NEXT UNPROTEC	
8391	1F96	.	.	*****	
8392	1F96	.	.	; SET LSTCOL PAST END OF LINE *	
8393	1F96	.	.	; TO CAUSE LINE TO BE RESCANNED *	
8394	1F96	.	.	*****	
8395	1F96	.	.	FLDSRX EQU \$	
8396	1F96	.	.	FSR140 EQU \$;(Z TRUE)	
8397	1F96	21	C8 FF	LXI H,LSTCOL	
8398	1F99	36	50	MVI M,MAXCOL+1	
8399	1F9B	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 258
=====
8401      1F9C      . . .      ;*****
8402      1F9C      . . .      ; UNPROTECTED FIELD FOUND *
8403      1F9C      . . .      ; SET NEW CURSOR POSITION *
8404      1F9C      . . .      ;*****
8405      1F9C      . . .      FSR200 EQU $
8406      1F9C      3E 50 .      MVI A,MAXCOL+1 ;COMPUTE NEW COLUMN
8407      1F9E      91 . .      SUB C
8408      1F9F      CD 7F 23      CALL CRRET1    ;SET CURRENT CURSOR LOCATION
8409      1FA2      32 C8 FF      STA LSTCOL    ;AND LAST CURSOR VALUE
8410      1FA5      EB . .      XCHG          ;STORE NEW CURRENT ADDRESS
8411      1FA6      22 C3 FF      SHLD CURADR
8412      1FA9      22 D5 FF      SHLD LADDR    ;SAVE FIELD ADDRESS IN
8413      1FAC      . . .      ;          CASE ROLL UP NEEDED
8414      1FAC      EB . .      XCHG          ;RESTORE D,E AND H,L
8415      1FAD      . . .      ;*****
8416      1FAD      . . .      ; COMPUTE NEW CURSOR ROW *
8417      1FAU      . . .      ;*****
8418      1FAD      3A DA FF      LDA NEWROW    ;GET NEW ABSOLUTE ROW NUMBER
8419      1FB0      B7 . .      ORA A         ;HAS ROW CHANGED?
8420      1FB1      CA F1 1F      JZ FSR360     ;NO - RETURN
8421      1FB4      21 C0 FF      LXI H,CURROW ;YES - CALCULATE NEW
8422      1FB7      86 . .      ADD M         ;ROW NUMBER
8423      1FB8      . . .      FSR240 EQU $
8424      1FB8      0E 18 .      MVI C,MAXROW+1 ;IS NEW ROW ON CURRENT PAGE?
8425      1FBA      B9 . .      CMP C
8426      1FBB      DA DA 1F      JC FSR340    ;YES
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 259
8428	1FBE	.	.	*****	
8429	1FBE	.	.	; NEW CURSOR ROW IS ON NEW PAGE *	
8430	1FBE	.	.	; ROLL DISPLAY UP TO GET ROW ON SCREEN *	
8431	1FBE	.	.	*****	
8432	1FBE	91	.	SUB C ;DECREMENT ROLL COUNT BY ONE	
8433	1FBF	21	6B	FF LXI H,MLKROW ;PAGE	
8434	1FC2	86	.	ADD M ;ADJUST FOR LOCKED DISPLAY	
8435	1FC3	57	.	MOV D,A ;SAVE RESULT FOR STORAGE	
8436	1FC4	79	.	MOV A,C ;COMPUTE NUMBER OF LINES TO	
8437	1FC5	96	.	SUB M ;ROLL FOR ONE PAGE	
8438	1FC6	5F	.	MOV E,A ;SAVE THE VALUE FOR STORAGE	
8439	1FC7	EB	.	XCHG ;PUT VALUES INTO H,L	
8440	1FC8	22	82	FF SHLD ROLLCT ;STORE ROLL PARAMETERS	
8441	1FCB	.	.	;	
8442	1FCB	.	.	; ROLL UP ONE PAGE OF LINES	
8443	1FCB	.	.	;	
8444	1FCB	.	.	FSR300 EQU \$	
8445	1FCB	CD	30	0D CALL ROLLUP ;ROLLUP ONE LINE	
8446	1FCE	21	82	FF LXI H,ROLLCT	
8447	1FD1	35	.	DCR M ;PAGE ROLLED UP?	
8448	1FD2	C2	CB	1F JNZ FSR300 ;NO - DO ANOTHER LINE	
8449	1FD5	23	.	INX H ;YES - GET NUMBER OF ROWS	
8450	1FD6	7E	.	MOV A,M ;TO UNPROTECTED FIELD AND	
8451	1FD7	C3	88	1F JMP FSR240 ;CHECK TO SEE IF ON SCREEN	
8452	1FDA	.	.	*****	
8453	1FDA	.	.	; UPDATE ROW *	
8454	1FDA	.	.	*****	
8455	1FDA	.	.	FSR340 EQU \$	
8456	1FDA	32	C0	FF STA CURROW ;SET NEW ROW NUMBER	
8457	1FDD	2A	C0	FF LHLD CURROW ;SET LAST ROW AND COLUMN DON	
8458	1FE0	22	C7	FF SHLD LSTROW ;CURRENT ROW AND COLUMN	
8459	1FE3	2A	96	FF LHLD LNKSAV ;SET "LSTLIN" TO CURRENT ROW	
8460	1FE6	23	.	INX H ;ADDRESS	
8461	1FE7	22	C9	FF SHLD LSTLIN	
8462	1FEA	2A	D5	FF LHLD LADDR ;SET "CURADR" TO ADDRESS OF	
8463	1FED	22	C3	FF SHLD CURADR ;FIRST CHAR IN NEW FIELD	
8464	1FF0	EB	.	XCHG ;PUT CURRENT ADDRESS INTO D,	
8465	1FF1	.	.	FSR360 EQU \$	
8466	1FF1	FE	44	. CPI D ;SET Z-FALSE (D >= 320)	
8467	1FF3	C3	D3	10 JMP DISLN1 ;GO SET DISPLAY CURSOR ROW	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 260
=====
8469      1FF6      . . .      ;
8470      1FF6      . . .      ; * * * * *
8471      1FF6      . . .      ;
8472      1FF6      . . .      ; FLDSR2 - DETERMINE PROTECT SENSE OF NEXT
8473      1FF6      . . .      ; CHARACTER
8474      1FF6      . . .      ;
8475      1FF6      . . .      ; ENTRY: D,E = NEXT CHARACTER ADDRESS
8476      1FF6      . . .      ;
8477      1FF6      . . .      ; EXIT : Z - CONTINUATION OF FORMAT FIELD
8478      1FF6      . . .      ; NZ - NOT A CONTINUATION
8479      1FF6      . . .      ; D,E = ADDRESS OF CHARACTER
8480      1FF6      . . .      ; H = BASEH
8481      1FF6      . . .      ; TEMP = NEW ENHANCEMENT CODE IF ANY
8482      1FF6      . . .      ; A,L DESTROYED
8483      1FF6      . . .      ;
8484      1FF6      . . .      FS2000 EQU $
8485      1FF6      32 9D FF      STA TEMP ;STORE NEW DISPLAY CONTROL
8486      1FF9      . . .      FLDSRB EQU $
8487      1FF9      . . .      FS2005 EQU $
8488      1FF9      1B . .      DCX D ;SET ADDRESS TO NEXT CHAR
8489      1FFA      . . .      FLDSR2 EQU $
8490      1FFA      . . .      ;*****
8491      1FFA      . . .      ; ROM BREAK 4
8492      1FFA      C3 02 20      JMP ZBRK4C
8493      1FFD      . . .      ORG ZBRK3+40000
8494      2000      . . .      ZBRK4 EQU $
8495      2000      54 . .      DB VERSN ;ROM PRESENT FLAGS
8496      2001      20 . .      DB ZBRK4/256
8497      2002      . . .      ZBRK4C EQU $
8498      2002      . . .      ;*****
8499      2002      13 . .      INX D ;SET ADDRESS TO PREV CHAR
8500      2003      CD 90 0C      CALL NXTCHR ;GET NEXT CHARACTER
8501      2006      C2 FA 1F      JNZ FLDSR2 ;SKIP OVER LINKS
8502      2009      87 . .      ADD A ;ASCII OR DISPLAY CONTROL?
8503      200A      D2 0A 0C      JNC NZEXIT ;ASCII - RETURN NOT CONTINUE
8504      200D      1F . .      RAR ;(RESTORE DATA BYTE)
8505      200E      F2 F6 1F      JP FS2000 ;DISPLAY CONTROL - IGNORE IT
8506      2011      FE C4 .      CPI STPFLG ;TERMINATOR OR TYPE DEFINE?
8507      2013      F2 F9 1F      JP FS2005 ;YES - SKIP TO NEXT CHARACTE
8508      2016      21 C5 FF      LXI H,LSTFMT ;COMPARE AGAINST LAST FORMAT
8509      2019      BE . .      CMP M ;CONTROL AND RETURN
8510      201A      C9 . .      RET
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 261
8512	201B	.	.	.	;
8513	201B	.	.	.	; * * * * *
8514	201B	.	.	.	;
8515	201B	.	.	.	; FNDCH - SEE IF NEXT CHAR IS FORMAT CONTROL BYTE
8516	201B	.	.	.	;
8517	201B	.	.	.	; ENTRY: TERMINAL IS IN FORMAT MODE
8518	201B	.	.	.	; D,E = START ADDRESS
8519	201B	.	.	.	; H,L = CHARACTERS TO LOOK FOR
8520	201B	.	.	.	;
8521	201B	.	.	.	; EXIT : Z - CHARACTER NOT FOUND
8522	201B	.	.	.	; NZ - CHARACTER FOUND
8523	201B	.	.	.	; D,E = ADDRESS OF ENDING CHARACTER
8524	201B	.	.	.	; A,B,C,L,TEMP DESTROYED
8525	201B	.	.	.	;
8526	201B	.	.	.	; FNDCH0 - SEE IF NEXT CHARACTER IS PROTECTED
8527	201B	.	.	.	;
8528	201B	.	.	.	FNDCH0 EQU \$
8529	201B	21	C0	C0	LXI H,STPR*256+STPR ;SET COMPARE CHARS
8530	201E	.	.	.	FNDCH EQU \$
8531	201E	3E	01	.	MVI A,IGNTRM ;SET TO IGNORE NON-DISPLAYIN
8532	2020	32	6D	FF	STA TRMFCT ;TERMINATOR
8533	2023	3A	C2	FF	LDA PROFLD ;SAVE PROTECTED FIELD
8534	2026	F5	.	.	PUSH PSW ;STATUS
8535	2027	0E	00	.	MVI C,0 ;SET FOR NEXT CHARACTER ONLY
8536	2029	CD	46	20	CALL FCR400 ;LOCATE THE NEXT CHARACTER
8537	202C	3E	00	.	MVI A,DELTRM ;RESTORE FLAG TO DELETE NON-
8538	202E	32	6D	FF	STA TRMFCT ;DISPLAYING TERMINATOR
8539	2031	C1	.	.	POP B ;RESET PROTECT STATUS TO BE
8540	2032	78	.	.	MOV A,B ;CONSISTENT WITH CHARACTER
8541	2033	32	C2	FF	STA PROFLD ;POINTED TO BY "CURADR"
8542	2036	C9	.	.	RET ;RETURN

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 262
8544	2037	.	.	;	
8545	2037	.	.	; * * * * *	
8546	2037	.	.	;	
8547	2037	.	.	; FNDCHU - LOCATE NEXT UNPROTECTED FIELD	
8548	2037	.	.	; CONTROL BYTE IN CURRENT LINE	
8549	2037	.	.	;	
8550	2037	.	.	; ENTRY: TERMINAL IS IN FORMAT MODE	
8551	2037	.	.	; B = DON'T CARE	
8552	2037	.	.	; C = CURRENT COLUMN NUMBER	
8553	2037	.	.	; D,E = START ADDRESS	
8554	2037	.	.	;	
8555	2037	.	.	; EXIT : Z - CHARACTER NOT FOUND	
8556	2037	.	.	; NZ - CHARACTER FOUND	
8557	2037	.	.	; C = NUMBER OF CHARS TO END OF LINE	
8558	2037	.	.	; D,E = ADDRESS OF ENDING CHARACTER	
8559	2037	.	.	; PROFLD SET AS DEFINED	
8560	2037	.	.	; A,B,L DESTROYED	
8561	2037	.	.	;	
8562	2037	.	.	FNDCHU EQU \$	
8563	2037	CD	D7 13	CALL DCXB2D ;DATA COMM OR I/O BUFF INPUT	
8564	203A	21	C1 C1	LXI H,ENDPR*256+ENDPR ;(SET "ENDPR" ONLY)	
8565	203D	CA	42 20	JZ FNDCU1 ;NO - SKIP XMIT ONLY FIELDS	
8566	2040	2E	C2 .	MVI L,XMONLY ;YES - LOOK FOR "XMONLY" ALS	
8567	2042	.	.	;	
8568	2042	.	.	; LOCATE THE FORMAT CONTROL CHARACTER	
8569	2042	.	.	;	
8570	2042	.	.	FNDCU1 EQU \$	
8571	2042	3E	4F .	MVI A,MAXCOL ;COMPUTE NO. OF CHARS	
8572	2044	91	.	SUB C ;TO SEARCH	
8573	2045	4F	.	MOV C,A	
8574	2046	.	.	FCR400 EQU \$	
8575	2046	CD	4D 20	CALL FNDCHR ;LOOK FOR SPECIFIED CHARS	
8576	2049	C8	.	RZ ;RETURN IF EOL ENCOUNTERED	
8577	204A	AF	.	XRA A ;OTHERWISE, SET FLAG TO	
8578	204B	B1	.	ORA C ;SHOW IF CHARACTER FOUND	
8579	204C	C9	.	RET	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS
8581	204D	. . .	;*****
8582	204D	. . .	; FNDCHR - LOCATE SPECIFIED CHARACTER *
8583	204D	. . .	;*****
8584	204D	. . .	;
8585	204D	. . .	; ENTRY: C = NUMBER OF COLUMNS TO SEARCH
8586	204D	. . .	; D,E = STARTING ADDRESS
8587	204D	. . .	; H,L = CHARACTERS TO LOOK FOR
8588	204D	. . .	; (VALID FOR FORMAT MODE ONLY)
8589	204D	. . .	;
8590	204D	. . .	; EXIT : Z - CHARACTER NOT FOUND
8591	204D	. . .	; NZ - CHARACTER FOUND
8592	204D	. . .	; C = NUMBER OF CHARACTERS LEFT
8593	204D	. . .	; (= 0, IF CHARACTER FOUND)
8594	204D	. . .	; D,E = ADDRESS OF TERMINATING CHARACTER
8595	204D	. . .	; "EOLMV" SET TO ZERO
8596	204D	. . .	; "PROFLD" SET IF IN FORMAT MODE
8597	204D	. . .	; "LSTFMT" UPDATED IF A FORMAT CONTROL
8598	204D	. . .	; CHARACTER IS ENCOUNTERED
8599	204D	. . .	;
8600	204D	. . .	FNDCHR EQU \$
8601	204D	AF . .	XRA A
8602	204E	32 90 FF	STA EOLMV
8603	2051	13 . .	INX D ;SET TO PREV CHAR ADDRESS
8604	2052	0C . .	INR C ;ADJUST CHARACTER COUNT
8605	2053	0C . .	INR C
8606	2054	. . .	FCR005 EQU \$
8607	2054	0D . .	DCR C ;COLUMN FOUND?
8608	2055	CA 0A 0C	JZ NZEXIT ;YES - RETURN CHARACTER FOUN
8609	2058	. . .	;
8610	2058	. . .	; SEARCH DISPLAY LIST
8611	2058	. . .	;
8612	2058	. . .	FCR010 EQU \$
8613	2058	CD 90 0C	CALL NXTCHR ;GET THE NEXT CHARACTER
8614	205B	C2 D4 20	JNZ FCR260 ;EOL LINK - EXIT NOT FOUND
8615	205E	87 . .	ADD A ;IS IT ASCII?
8616	205F	D2 54 20	JNC FCR005 ;YES - DECREMENT COLUMN COUN
8617	2062	. . .	;*****
8618	2062	. . .	; NON-ASCII CHARACTER - DETERMINE CHAR FUNCTION *
8619	2062	. . .	;*****
8620	2062	1F . .	RAR ;RESTORE CHARACTER
8621	2063	FA 6C 20	JM FCR100 ;NOT DISPLAY CTL - CHECK MOR
8622	2066	32 C6 FF	STA LSTOCD ;UPDATE CURRENT DISPLAY CODE
8623	2069	C3 58 20	JMP FCR010 ;CONTINUE SEARCHING

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 264
=====
8625      206C      . . .      ;
8626      206C      . . .      ; FORMAT CONTROL CHARACTER - CHECK FOR ENDING
8627      206C      . . .      ;
8628      206C      . . .      FCR100 EQU $
8629      206C      FE CC .      CPI EOL          ;END OF LINE?
8630      206E      C8 . .      RZ              ;YES - RETURN
8631      206F      FE CE .      CPI EOP          ;END OF DISPLAY?
8632      2071      C8 . .      RZ              ;YES - RETURN
8633      2072      FE C4 .      CPI STPFLG      ;NON-DISPLAYING TERMINATOR?
8634      2074      CA C3 20     JZ FCR200       ;YES - DETERMINE ITS FUNCTIO
8635      2077      FE C5 .      CPI ALPHA       ;TYPE DEFINITION?
8636      2079      F2 A4 20     JP FCR150       ;YES - SET CHECK FUNCTION
8637      207C      FE C3 .      CPI XMONLY+1    ;FORMAT CONTROL?
8638      207E      F2 58 20     JP FCR010       ;NO - CONTINUE SEARCHING
8639      2081      E5 . .      PUSH H          ;YES - RESET CHECK ROUTINE
8640      2082      21 08 08     LXI H,ZRETRN   ;ADDRESS
8641      2085      22 86 FF     SHLD CHKRTN
8642      2088      E1 . .      POP H           ;RESTORE CHECK CHARACTERS
8643      2089      32 C5 FF     STA LSTFMT     ;SET CURRENT FORMAT CONTROL
8644      208C      47 . .      MOV B,A        ;SAVE CONTROL CHARACTER
8645      208D      CD CF 1A     CALL CHKFMS    ;FORMAT/SOFT KEY DEFINE MODE
8646      2090      CA 58 20     JZ FCR010       ;NO - CONTINUE SEARCHING
8647      2093      78 . .      MOV A,B        ;RECALL CHARACTER
8648      2094      DE C1 .      SBI STPR+1     ;COMPUTE "PROFLD" VALUE
8649      2096      32 C2 FF     STA PROFLD     ;(= -1 FOR PROTECTED)
8650      2099      78 . .      MOV A,B        ;RECALL CHARACTER
8651      209A      BC . .      CMP H          ;TERMINATOR FOUND?
8652      209B      CA A2 20     JZ FCR110       ;YES - EXIT
8653      209E      BD . .      CMP L
8654      209F      C2 58 20     JNZ FCR010     ;NO - CONTINUE SEARCHING
8655      20A2      . . .      FCR110 EQU $
8656      20A2      B7 . .      ORA A          ;SET Z FALSE
8657      20A3      C9 . .      RET           ;RETURN
8658      20A4      . . .      ;
8659      20A4      . . .      ; TYPE DEFINITION FOUND - SET CHECK ROUTINE
8660      20A4      . . .      ;
8661      20A4      . . .      FCR150 EQU $
8662      20A4      E5 . .      PUSH H         ;SAVE TERMINATOR CHARACTERS
8663      20A5      21 23 48     LXI H,ZALPCK   ;SET H,L FOR ALPHA CHECK
8664      20A8      CA BC 20     JZ FCR160      ;SET ALPHA CHECK IF ALPHA
8665      20AB      21 26 48     LXI H,ZNUMCK   ;SET H,L FOR NUMERIC CHECK
8666      20AE      D6 C7 .      SUI NUMBER+1   ;NUMERIC FIELD?
8667      20B0      FA BC 20     JM FCR160      ;YES - SET CHECK ROUTINE ADD
8668      20B3      21 0B 08     LXI H,ZRETRN   ;NO - SET H,L FOR ALPHANUM
8669      20B6      CA BC 20     JZ FCR160      ;SET ROUTINE ADDR IF = ZERO
8670      20B9      21 EC 10     LXI H,SFKCHK   ;ELSE, SET FOR SOFT KEYS
8671      20BC      . . .      FCR160 EQU $
8672      20BC      22 86 FF     SHLD CHKRTN   ;SET CHECK ROUTINE ADDRESS
8673      20BF      E1 . .      POP H         ;RECALL TERMINATOR CHARACTER
8674      20C0      C3 58 20     JMP FCR010     ;CONTINUE SEARCHING
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 265
=====
8676     20C3      . . .      ;*****
8677     20C3      . . .      ; NON-DISPLAYING TERMINATOR FOUND - DETERMINE *
8678     20C3      . . .      ;   AND PERFORM ITS FUNCTION                       *
8679     20C3      . . .      ;*****
8680     20C3      . . .      FCR200 EQU $
8681     20C3      3A 6D FF   LDA TRMFACT ;GET THE FUNCTION FLAG
8682     20C6      B7 . .     ORA A       ;WHAT FUNCTION?
8683     20C7      FA 03 20   JM FCR250   ;-1 - TERMINATE TRANSFER
8684     20CA      C2 58 20   JNZ FCR010  ;+1 - IGNORE IT
8685     20CD      CD 11 1C   CALL CHRDL2 ;0 - DELETE IT
8686     20D0      C3 58 20   JMP FCR010  ;CONTINUE CHARACTER SEARCH
8687     20D3      . . .      ;
8688     20D3      . . .      ; TERMINATE TRANSFER
8689     20D3      . . .      ;
8690     20D3      . . .      FCR250 EQU $
8691     20D3      1A . .     LDAX D      ;PUT CHARACTER BACK IN A-REG
8692     20D4      . . .      FCR260 EQU $
8693     20D4      8F . .     CMP A       ;SET Z-FLAG TRUE
8694     20D5      C9 . .     RET        ;RETURN CHARACTER NOT FOUND
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 266
8696	20D6	.	.	*****	
8697	20D6	.	.	; FNDLST - LOCATE LAST CHARACTER TYPE AHEAD OF *	
8698	20D6	.	.	; CURRENT CHARACTER *	
8699	20D6	.	.	*****	
8700	20D6	.	.	;	
8701	20D6	.	.	; ENTRY: A = NUMBER OF COLUMNS TO SEARCH	
8702	20D6	.	.	; D,E = ADDRESS OF CHARACTER BEFORE	
8703	20D6	.	.	; BEFORE FIRST CHARACTER TO LOOK AT	
8704	20D6	.	.	; H,L = CHARACTERS TO BE FOUND	
8705	20D6	.	.	;	
8706	20D6	.	.	; EXIT: P - CHARACTER FOUND	
8707	20D6	.	.	; B = NUMBER OF CHARACTERS FROM CURRENT	
8708	20D6	.	.	; CHARACTER	
8709	20D6	.	.	; M - CHARACTER NOT FOUND	
8710	20D6	.	.	; B DESTROYED	
8711	20D6	.	.	; A,C,D,E DESTROYED	
8712	20D6	.	.	;	
8713	20D6	.	.	FNDLS0 EQU \$	
8714	20D6	3C	.	INR A ;ADJUST SEARCH COUNT	
8715	20D7	21	CC CC	LXI H,EOL*256+EOL ;SET TO LOOK FOR "EOL"	
8716	20DA	.	.	;	
8717	20DA	.	.	FNDLST EQU \$	
8718	20DA	4F	.	MOV C,A ;PUT SEARCH COUNT IN C-REG	
8719	20DB	06	FF .	MVI B,377Q ;PRESET B FOR FAIL RETURN	
8720	20DD	3D	.	DCR A ;ANY COLUMNS TO SEARCH?	
8721	20DE	F8	.	RM ;NO - RETURN NONE FOUND	
8722	20DF	.	.	FLS010 EQU \$	
8723	20DF	CD	90 0C	CALL NXTCHR ;GET THE NEXT CHARACTER	
8724	20E2	BC	.	CMP H ;DOES IT MATCH DESIRED CHARS	
8725	20E3	CA	EA 20	JZ FLS020 ;YES - SAVE LOCATION OF CHAR	
8726	20E6	BD	.	CMP L	
8727	20E7	C2	EB 20	JNZ FLS030 ;NO - GO TO NEXT CHARACTER	
8728	20EA	.	.	FLS020 EQU \$	
8729	20EA	41	.	MOV B,C ;SAVE LOCATION OF CHAR IN B	
8730	20EB	.	.	FLS030 EQU \$	
8731	20EB	B7	.	ORA A ;IS CURRENT CHAR ASCII?	
8732	20EC	FA	F6 20	JM FLS050 ;NO - CHECK FOR TERMINATION	
8733	20EF	0D	.	DCR C ;SEARCH COMPLETE?	
8734	20F0	.	.	FLS035 EQU \$	
8735	20F0	C2	DF 20	JNZ FLS010 ;NO - CHECK NEXT CHARACTER	
8736	20F3	.	.	FLS040 EQU \$	
8737	20F3	AF	.	XRA A ;CLEAR A-REGISTER	
8738	20F4	B0	.	ORA B ;SET FLAGS FOR RETURN	
8739	20F5	C9	.	RET ;RETURN	
8740	20F6	.	.	*****	
8741	20F6	.	.	; NON-ASCII CHARACTER - CHECK FOR TERMINATION *	
8742	20F6	.	.	*****	
8743	20F6	.	.	FLS050 EQU \$	
8744	20F6	FE	CC .	CPI EOL ;IS IT AN EOL?	
8745	20F8	CA	F3 20	JZ FLS040 ;YES - EXIT	

13255
2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
8746     20FB     FE CE      .          CPI EOP      ;IS IT AN EOP?
8747     20FD     C3 F0      20        JMP FLS035   ;GO CHECK RESULT
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 268
8749	2100	.	.	;*****	
8750	2100	.	.	; HTAB - SKIP TO NEXT TAB POSITION *	
8751	2100	.	.	;*****	
8752	2100	.	.	HTAB EQU \$	
8753	2100	.	.	;***** GRAPHICS MODIFICATION *****	
8754	2100	3A	97 90	LDA ZGFLG6 ;PROCESS IN A/N?	
8755	2103	E6	02 .	ANI GTEXT	
8756	2105	C4	41 60	CNZ ZHT	
8757	2108	D8	.	RC ;NO, PROCESS IN GRAPHICS	
8758	2109	.	.	;*****	
8759	2109	CD	CB 1A	CALL CHKFM0 ;FORMAT/SOFT KEY DEFINE MODE	
8760	210C	C2	55 21	JNZ HTB200 ;YES - LOCATE NEXT FIELD	
8761	210F	2E	C1 .	MVI L,CURCOL-BASE ;NO - LOCATE NEXT TAB	
8762	2111	46	.	MOV B,M ;SET POSITION	
8763	2112	04	.	INR B ;START FROM NEXT COLUMN	
8764	2113	3E	4F .	MVI A,MAXCOL ;COMPUTE NUMBER OF COLUMNS	
8765	2115	90	.	SUB B ;TO END OF LINE	
8766	2116	FA	22 22	JM CRLF ;GO TO START OF NEXT LINE IF	
8767	2119	.	.	; ALREADY AT END OF LINE	
8768	2119	F6	07 .	ORI 7 ;MOVE TO COL CORRESP. TO	
8769	211B	.	.	; START OF BYTE	
8770	211B	4F	.	MOV C,A ;SAVE IN C	
8771	211C	78	.	MOV A,B	
8772	211D	CD	3F 16	CALL FNDBT1 ;GET TABLE ENTRY FOR COLUMN	
8773	2120	3D	.	DCR A ;MASK OFF BITS FOR	
8774	2121	2F	.	CMA ;PREVIOUS COLUMNS	
8775	2122	A6	.	ANA M	
8776	2123	.	.	;*****	
8777	2123	.	.	; CHECK NEXT COLUMN FOR SET TAB *	
8778	2123	.	.	;*****	
8779	2123	.	.	HTB100 EQU \$	
8780	2123	06	08 .	MVI B,8 ;GET BIT COUNT	
8781	2125	CA	48 21	JZ HTB140 ;NO BITS SET IN BYTE	
8782	2128	.	.	HTB120 EQU \$	
8783	2128	0F	.	RRC ;TAB BIT SET?	
8784	2129	D2	44 21	JNC HTB130 ;NO - TRY NEXT COLUMN	
8785	212C	.	.	;*****	
8786	212C	.	.	; TAB IS SET - UPDATE CURCOL *	
8787	212C	.	.	;*****	
8788	212C	.	.	HTB160 EQU \$	
8789	212C	5F	.	MOV E,A ;SAVE A-REGISTER	
8790	212D	3E	56 .	MVI A,MAXCOL+7 ;COMPUTE COLUMN OF LOCATIO	
8791	212F	91	.	SUB C ;OF TAB	
8792	2130	90	.	SUB B	
8793	2131	22	96 FF	SHLD LNKSAV ;SAVE CURRENT TABLE ADDRESS	
8794	2134	2A	BE FF	LHLD RHTMGN ;GET RIGHT AND LEFT MARGINS	
8795	2137	BD	.	CMP L ;TAB BEYOND RIGHT MARGIN?	
8796	2138	F2	22 22	JP CRLF ;YES - DO CR, LF	
8797	213B	3C	.	INR A ;NO - ADJUST TO PROPER VALUE	
8798	213C	BC	.	CMP H ;TAB BEYOND LEFT MARGIN?	

13255
2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 269
=====
8799     213D     D2 C9 12      JNC  CURP04      ;YES - LOCATE TAB LOCATION
8800     2140     7B . .        MOV  A,E         ;NO - RESTORE A-REGISTER
8801     2141     2A 96 FF      LHLD LNKSAV     ;RECALL TAB TABLE ADDRESS
8802     2144     . . . ;        ;                LOOK FOR ANOTHER TAB
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 270
=====
8804      2144      . . .      ;*****
8805      2144      . . .      ; TAB NOT FOUND - CHECK NEXT COLUMN *
8806      2144      . . .      ;*****
8807      2144      . . .      HTB130 EQU $          ;NO - TRY NEXT COLUMN
8808      2144      05 . .      DCR B                ;ALL BITS EXAMINED?
8809      2145      C2 28 21      JNZ HTB120           ;NO - LOOK TO NEXT BIT
8810      2148      . . .      ;*****
8811      2148      . . .      ; BYTE EXHAUSTED *
8812      2148      . . .      ; MOVE TO NEXT TABTBL ENTRY *
8813      2148      . . .      ;*****
8814      2148      . . .      HTB140 EQU $
8815      2148      79 . .      MOV A,C              ;GET COLUMN COUNT
8816      2149      D6 08 .      SUI 8                ;DECREMENT
8817      2148      FA 22 22      JM CRLF              ;DO CR,LF IF REACHED END
8818      214E      4F . .      MOV C,A
8819      214F      23 . .      INX H                ;GET NEXT BYTE FROM TABLE
8820      2150      7E . .      MOV A,M
8821      2151      B7 . .      ORA A                ;SET FLAGS
8822      2152      C3 23 21      JMP HTB100
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 271
=====
8824      2155      . . .      ;*****
8825      2155      . . .      ; FORMAT MODE TAB *
8826      2155      . . .      ;*****
8827      2155      . . .      HTB200 EQU $
8828      2155      CD 3A 1F    CALL FLDSR      ;SEARCH FOR NEXT FIELD
8829      2158      C0 . .      RNZ              ;RETURN IF FOUND
8830      2159      C3 A2 1E    JMP CURPH1      ;HOME TO FIRST UNPROT. FIELD
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 272
=====
8832      215C      . . .      ;
8833      215C      . . .      ; * * * * *
8834      215C      . . .      ;
8835      215C      . . .      ; ICHON,ICHOFF - INSERT CHARACTER ON/OFF
8836      215C      . . .      ;
8837      215C      . . .      ICHON EQU $
8838      215C      06 00      MVI B,0          ;SET FOR NO BLINK
8839      215E      . . .      ICH010 EQU $
8840      215E      3E 02      MVI A,INSCHR    ;TURN ON INSERT CHARACTER
8841      2160      C3 0E 48    JMP ZSTMD1      ;LED AND EXIT
8842      2163      . . .      ;
8843      2163      . . .      ICHOFF EQU $
8844      2163      3E FD      MVI A,377Q-INSWRP
8845      2165      CD 20 15    CALL CLCMFL     ;CLEAR WRAP AROUND FLAG
8846      2168      3E 02      MVI A,INSCHR    ;TURN OFF INSERT CHARACTER
8847      216A      C3 11 48    JMP ZCLMD1
8848      216D      . . .      ;*****
8849      216D      . . .      ; IWRPON - INSERT WITH WRAPAROUND ON *
8850      216D      . . .      ;*****
8851      216D      . . .      IWRPON EQU $
8852      216D      3E 02      MVI A,INSWRP
8853      216F      CD 44 15    CALL STCMFL     ;SET WRAP AROUND FLAG
8854      2172      06 FF      MVI B,377Q     ;SET TO BLINK LED
8855      2174      C3 5E 21    JMP ICH010     ;SET INSERT CHARACTER LED ON
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
8857	2177	.	.	*****
8858	2177	.	.	; BCKSPC - BACKSPACE ONE CHARACTER POSITION *
8859	2177	.	.	*****
8860	2177	.	.	BCKSPC EQU \$
8861	2177	.	.	***** GRAPHICS MODIFICATION *****
8862	2177	3A	97 90	LDA ZGFLG6 ;PROCESS IN A/N?
8863	217A	E6	02 .	ANI GTEXT
8864	217C	C4	47 60	CNZ ZBS
8865	217F	D8	. .	RC ;NO, PROCESS IN GRAPHICS
8866	2180	.	.	*****
8867	2180	2E	C1 .	MVI L,CURCOL-BASE
8868	2182	35	. .	DCR M ;DECREMENT CURRENT COLUMN
8869	2183	F0	. .	RP ;RETURN IF NOT AT COLUMN ZER
8870	2184	34	. .	INR M ;ELSE, RESTORE TO ZERO AND
8871	2185	C9	. .	RET ;RETURN

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 274
8873	2186	.	.	*****	
8874	2186	.	.	; CURADV - CURSOR ADVANCE ROUTINE *	
8875	2186	.	.	; ADVANCES CURSOR TO NEXT POSITION *	
8876	2186	.	.	; ON DISPLAY *	
8877	2186	.	.	*****	
8878	2186	.	.	CURADV EQU \$;ADVANCE CURSOR TWICE	
8879	2186	CD	89 21	CALL CURADV ;DO FIRST CURSOR ADVANCE	
8880	2189	.	.	; THEN FALL IN TO DO NEXT	
8881	2189	.	.	CURADV EQU \$	
8882	2189	.	.	***** GRAPHICS MODIFICATION *****	
8883	2189	3A	96 FB	LDA ZAPFLG ;AUTO PLOT ON?	
8884	218C	E6	02 .	ANI APIP	
8885	218E	C4	10 60	CNZ ZAPSCN ;IF YES, DO SCAN	
8886	2191	.	.	*****	
8887	2191	CD	E3 21	CALL CRADV ;ADVANCE CURSOR	
8888	2194	CD	CF 1A	CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE	
8889	2197	C8	.	RZ ;NO - RETURN	
8890	2198	.	.	*****	
8891	2198	.	.	; FORMAT MODE *	
8892	2198	.	.	; CHECK FOR ADVANCE INTO PROTECTED FIELD *	
8893	2198	.	.	*****	
8894	2198	3A	C1 FF	LDA CURCOL ;GET NEW CURRENT COLUMN	
8895	219B	B7	.	ORA A ;DID CURSOR WRAP AROUND?	
8896	219C	C2	B4 21	JNZ CRA040 ;NO - CHECK FOR PROTECTED FL	
8897	219F	.	.	*****	
8898	219F	.	.	; CURSOR WRAPPED AROUND *	
8899	219F	.	.	; SEE IF NEW LINE IS CONTINUATION *	
8900	219F	.	.	; OF UNPROTECTED FIELD *	
8901	219F	.	.	*****	
8902	219F	2A	C9 FF	LHLD LSTLIN	
8903	21A2	EB	.	XCHG ;GET CURRENT LINE ADDR IN D,	
8904	21A3	3A	8A FF	LDA FMTCTL ;RESET "LSTFMT" TO LAST	
8905	21A6	32	C5 FF	STA LSTFMT ;FORMAT CONTROL IN LINE	
8906	21A9	CD	F9 1F	CALL FLDSRB ;CONTINUATION FIELD?	
8907	21AC	C2	C5 21	JNZ CRA060 ;NO - TAB TO NEXT FIELD	
8908	21AF	.	.	*****	
8909	21AF	.	.	; RESET CURADV FLAG *	
8910	21AF	.	.	*****	
8911	21AF	.	.	CRADV1 EQU \$	
8912	21AF	AF	.	XRA A	
8913	21B0	32	67 FF	STA CRAFLG	
8914	21B3	C9	.	RET	

13255
2648A MICROCODE LISTING 'PT91'

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
8916	21B4	.	.	;*****
8917	21B4	.	.	; CURSOR DID NOT WRAP AROUND *
8918	21B4	.	.	; SEE IF CURSOR ENTERED *
8919	21B4	.	.	; PROTECTED FIELD *
8920	21B4	.	.	;*****
8921	21B4	.	.	CRA040 EQU \$
8922	21B4	2A	C3 FF	LHLD CURADR ;GET THE CURRENT CHAR ADDR
8923	21B7	EB	.	XCHG ;PUT IT INTO H,L
8924	21B8	1B	.	DCX D ;SET POINTER TO NEXT CHAR
8925	21B9	2A	86 FF	LHLD CHKRTN ;SAVE THE CURRENT CHECK
8926	21BC	E5	.	PUSH H ;ROUTINE ADDRESS
8927	21BD	CD	1B 20	CALL FNDCHU ;NEXT CHARACTER PROTECTED?
8928	21C0	E1	.	POP H ;(RESTORE CHECK ROUTINE
8929	21C1	22	86 FF	SHLD CHKRTN ;ROUTINE ADDRESS)
8930	21C4	C8	.	RZ ;NO - RETURN
8931	21C5	.	.	CRA060 EQU \$
8932	21C5	CD	AF 21	CALL CRA0V1 ;RESET CURADV FLAG
8933	21C8	CD	D7 13	CALL DCXB2D ;DATA COMM OR I/O BUFF CHAR?
8934	21CB	21	C2 C1	LXI H,ENDPR*256+XMONLY ;(SET DEFAULT)
8935	21CE	C2	D5 21	JNZ CRA070 ;YES - DON'T SOUND BELL
8936	21D1	CD	14 48	CALL ZBELL ;NO - SOUND BELL
8937	21D4	6C	.	MOV L,H ;LOOK FOR "ENDPR" ONLY
8938	21D5	.	.	CRA070 EQU \$
8939	21D5	CD	1E 20	CALL FNDCH ;NEXT CHARACTER UNPROTECTED
8940	21D8	CC	3A 1F	CZ FLDSR ;OR ANOTHER FIELD EXIST?
8941	21DB	C0	.	RNZ ;YES - RETURN
8942	21DC	CD	D7 13	CALL DCXB2D ;DATA FROM DATA COMM OR CTU?
8943	21DF	C8	.	RZ ;NO, FROM KEYBOARD - RETURN
8944	21E0	C3	A2 1E	JMP CURPH1 ;YES - HOME THE CURSOR

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 276
8946	21E3	.	.	*****	
8947	21E3	.	.	; CRADV - ADVANCE CURSOR *	
8948	21E3	.	.	*****	
8949	21E3	.	.	CRADV EQU \$	
8950	21E3	3A	BE FF	LDA RHTMGN ;GET RIGHT MARGIN SETTING	
8951	21E6	21	C1 FF	LXI H,CURCOL	
8952	21E9	.	.	CRA010 EQU \$	
8953	21E9	BE	.	CMP M ;CURSOR AT RIGHT MARGIN?	
8954	21EA	CA	03 22	JZ CRA100 ;YES - CHECK FOR WRAP AROUND	
8955	21ED	3E	4F .	MVI A,MAXCOL ;(SET FOR LAST COL CHECK)	
8956	21EF	FA	E9 21	JM CRA010 ;AFTER MARGIN - CHECK EOL	
8957	21F2	34	.	INR M ;ADVANCE CURSOR	
8958	21F3	BE	.	CMP M ;MOVED INTO RIGHT MARGIN OR	
8959	21F4	C4	90 11	CNZ CKPROT ;INTO PROTECTED FIELD?	
8960	21F7	C8	.	RZ ;YES - DON'T SET CURADV FLAG	
8961	21F8	3A	F4 FF	LDA MDFLG1 ;GET TERMINAL MODE FLAGS	
8962	21FB	E6	02 .	ANI INSCHR ;IN CHARACTER INSERT MODE?	
8963	21FD	C0	.	RNZ ;YES - DON'T SET FLAG	
8964	21FE	2E	67 .	MVI L,CRAFLG-BASE ;NO - SET CURADV FLAG	
8965	2200	36	01 .	MVI M,1	
8966	2202	C9	.	RET	
8967	2203	.	.	*****	
8968	2203	.	.	; CURSOR IS IN LAST COLUMN OF LINE *	
8969	2203	.	.	*****	
8970	2203	.	.	CRA100 EQU \$	
8971	2203	3A	C5 FF	LDA LSTFMT ;SAVE LAST FORMAT CONTROL	
8972	2206	32	8A FF	STA FMTCTL ;IN CURRENT LINE	
8973	2209	CD	CF 1A	CALL CHKFMS ;FORMAT/SOFT KEY DEFINE OR	
8974	220C	CC	72 11	CZ CKDSPF ;DISPLAY FUNCTIONS ENABLED	
8975	220F	C2	22 22	JNZ CRLF ;YES - DON'T CLEAR WRAP FLAG	
8976	2212	3A	FB FF	LDA KBJMPR ;NO - GET KEYBOARD JUMPERS 1	
8977	2215	E6	04 .	ANI LINWRP ;WRAP AROUND ENABLED?	
8978	2217	C0	.	RNZ ;NO - RETURN	
8979	2218	3A	6E FF	LDA DFLGS ;YES - GET DATA TRANSFER FLG	
8980	221B	E6	80 .	ANI XBF2DS ;I/O BUFFER TO DISPLAY?	
8981	221D	3E	BF .	MVI A,377Q-WRPFLG ;(SET CLEAR MASK)	
8982	221F	C4	66 05	CNZ CLRMF2 ;YES - CLEAR LINE WRAP FLAG	
8983	2222	.	.	*****	
8984	2222	.	.	; CURSOR SHOULD BE WRAPPED INTO NEXT LINE *	
8985	2222	.	.	; GENERATE CR,LF *	
8986	2222	.	.	*****	
8987	2222	.	.	CRLF EQU \$	
8988	2222	CD	66 23	CALL CRRET ;CARRIAGE RETURN	
8989	2225	C3	6F 0B	JMP LNFEED ;LINE FEED	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 277
8991	2228	.	.	*****	
8992	2228	.	.	; CURPR - CURSOR POINTER RIGHT *	
8993	2228	.	.	*****	
8994	2228	.	.	CURPR EQU \$	
8995	2228	.	.	*****	
8996	2228	3E	00	MVI A,RIGHT ;SEE IF GRAPHICS TEXT	
8997	222A	CD	73 60	CALL ZANCUR ;CURSOR MOVE	
8998	222D	D8	.	RC ;DONE IF SO	
8999	222E	.	.	*****	
9000	222E	3E	01	MVI A,1 ;GET INCREMENT RIGHT	
9001	2230	C3	3B 22	JMP CURPL1	
9002	2233	.	.	*****	
9003	2233	.	.	; CURPL - CURSOR POINTER LEFT *	
9004	2233	.	.	*****	
9005	2233	.	.	CURPL EQU \$	
9006	2233	.	.	*****	
9007	2233	3E	02	MVI A,LEFT ;SEE IF GRAPHICS TEXT CURSOR	
9008	2235	CD	73 60	CALL ZANCUR ;MOVE	
9009	2238	D8	.	RC ;DONE IF SO	
9010	2239	.	.	*****	
9011	2239	3E	FF	MVI A,-1 ;GET INCREMENT LEFT	
9012	223B	.	.	CURPL1 EQU \$	
9013	223B	2E	C1	MVI L,CURCOL ;GET CURSOR COLUMN	
9014	223D	86	.	ADD M ;ADD INCREMENT	
9015	223E	77	.	MOV M,A ;STORE NEW COLUMN ADDRESS	
9016	223F	FA	51 22	JM CURPL2 ;WRAPAROUND TO LEFT	
9017	2242	D6	50	SUI MAXCOL+1 ;WRAPAROUND TO RIGHT?	
9018	2244	C0	.	RNZ ;NO - RETURN	
9019	2245	77	.	MOV M,A ;YES - SET TO COLUMN ZERO	
9020	2246	.	.	*****	
9021	2246	.	.	; CURPD - CURSOR POINTER DOWN *	
9022	2246	.	.	*****	
9023	2246	.	.	CURPD EQU \$	
9024	2246	.	.	*****	
9025	2246	3E	01	MVI A,DOWN ;SEE IF GRAPHICS TEXT	
9026	2248	CD	73 60	CALL ZANCUR ;CURSOR MOVE	
9027	224B	D8	.	RC ;DONE IF SO	
9028	224C	.	.	*****	
9029	224C	3E	01	MVI A,1	
9030	224E	C3	5B 22	JMP CURPU1	
9031	2251	.	.	*****	
9032	2251	.	.	; CURSOR MOVED OFF LEFT OF SCREEN *	
9033	2251	.	.	; WRAPAROUND TO RIGHT AND UP *	
9034	2251	.	.	*****	
9035	2251	.	.	CURPL2 EQU \$	
9036	2251	36	4F	MVI M,MAXCOL ;PUT CURSOR AT LAST COLUMN	
9037	2253	.	.	*****	
9038	2253	.	.	; CURPU - CURSOR POINTER UP *	
9039	2253	.	.	*****	
9040	2253	.	.	CURPU EQU \$	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 278
=====
9041      2253      . . .      ;*****
9042      2253      3E 03 .      MVI A,UP      ;SEE IF GRAPHICS TEXT
9043      2255      CD 73 60     CALL ZANCUR   ;CURSOR MOVE
9044      2258      D8 . .      RC           ;DONE IF SO
9045      2259      . . .      ;*****
9046      2259      3E 17 .      MVI A,MAXROW
9047      2258      . . .      CURPU1 EQU $
9048      2258      2E C0 .      MVI L,CURROW ;GET CURSOR ROW
9049      225D      86 . .      ADD M        ;ADD DISPLACEMENT
9050      225E      77 . .      MOV M,A     ;STORE NEW ROW ADDRESS
9051      225F      D6 18 .      SUI MAXROW+1 ;ROW LIMIT EXCEEDED?
9052      2261      F8 . .      RM          ;NO - RETURN
9053      2262      77 . .      MOV M,A     ;YES - STORE ADJUSTED ROW
9054      2263      C9 . .      RET         ;RETURN
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS
9056	2264	. . .	;*****
9057	2264	. . .	; DFSFKY - DEFINE SOFT KEYS *
9058	2264	. . .	;*****
9059	2264	. . .	DFSFKY EQU \$
9060	2264	2E D9 .	MVI L,SCRNRW ;CLEAR SOFT KEY PARAMETERS
9061	2266	1E 03 .	MVI E,3 ;TO ZERO
9062	2268	CD 30 12	CALL CLRAL1
9063	226B	. . .	;*****
9064	226B	3E 01 .	MVI A,DEFKEY ;CLEAR 'DEFINITION IN
9065	226D	CD 03 24	CALL CLSKFL ;PROGRESS' FLAG
9066	2270	. . .	;*****
9067	2270	21 87 7F	LXI H,DFSTAB ;SET RANGE TABLE FOR SOFT KE
9068	2273	C3 32 05	JMP ESCAPA ;DEFINITION ESCAPE SEQUENC
9069	2276	. . .	; A - DEFINE ATTRIBUTE CODE
9070	2276	. . .	; 0 = NORMAL
9071	2276	. . .	; 1 = LOCAL ONLY
9072	2276	. . .	; 2 = TRANSMIT ONLY
9073	2276	. . .	; DFS100 EQU \$
9074	2276	. . .	MVI C,2 ;SET MAXIMUM VALUE AND
9075	2276	. . .	LXI D,PARM2 ;PARAMETER TO BE SET
9076	2276	. . .	JMP DFS220 ;SET PARAMETER AND EXIT
9077	2276	0E 02 .	; K - KEY NUMBER TO BE DEFINED
9078	2278	11 DA FF	; DFS110 EQU \$
9079	2278	C3 9B 22	MVI C,NMFCTK-1 ;SET MAXIMUM VALUE AND
9080	227E	. . .	LXI D,PARM3 ;PARAMETER TO BE SET
9081	227E	. . .	;*****
9082	227E	. . .	JMP DFS220 ;KEY CODE = 0 NOW ALLOWED
9083	227E	. . .	;*****
9084	227E	0E 08 .	; L - SET LENGTH OF INPUT
9085	2280	11 D9 FF	; DFS120 EQU \$
9086	2283	. . .	MVI C,MAXCOL ;SET MAXIMUM VALUE AND
9087	2283	C3 9B 22	LXI D,PARM1
9088	2286	. . .	FALL INTO EVALUATION ROUTINE
9089	2286	. . .	
9090	2286	. . .	
9091	2286	. . .	
9092	2286	. . .	
9093	2286	0E 4F .	
9094	2288	11 DB FF	
9095	228B	. . .	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 280
9097	228B	.	.	.	;
9098	228B	.	.	.	;
9099	228B	.	.	.	;
9100	228B	.	.	.	;
9101	228B	.	.	.	;
9102	228B	.	.	.	;
9103	228B	.	.	.	;
9104	228B	2A	DE	FF	DFS200 EQU \$;ENTRY FOR MIN VALUE = 1
9105	228E	2B	.	.	LHLD IODATA ;GET INPUT PARAMETER
9106	228F	7C	.	.	DCX H ;ADJUST PARAMETER TO ONE LES
9107	2290	BD	.	.	MOV A,H ;CHECK FOR ZERO PARAMETER
9108	2291	C2	98	22	CMP L ;DOES MSB=LSB?
9109	2294	3C	.	.	JNZ DFS210 ;NO - STORE ADJUST VALUE
9110	2295	CA	9B	22	INR A ;IS ADJUST VALUE -1
9111	2298	.	.	.	JZ DFS220 ;YES - DON'T STORE NEW VALUE
9112	2298	22	DE	FF	DFS210 EQU \$;NO - STORE ADJUSTED VALUE
9113	229B	.	.	.	DFS220 EQU \$
9114	229B	CD	3C	11	CALL CHKLI0 ;EVALUATE AND SET PARAMETERS
9115	229E	.	.	.	*****
9116	229E	3E	01	.	MVI A,DEFKEY ;SET 'DEFINITION IN PROGRESS'
9117	22A0	CD	FD	23	CALL STSKFL ;FLAG
9118	22A3	.	.	.	*****
9119	22A3	3A	88	FF	LDA CHAR ;RECALL INPUT CHARACTER
9120	22A6	E6	20	.	ANI 40H ;IS IT UPPER CASE?
9121	22A8	C2	3A	05	JNZ ESCAPB ;NO - CONTINUE ESCAPE SEQ
9122	22AB	.	.	.	YES - SET NEW DEFINITION

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
9124	22AB	.	.	. ;*****
9125	22AB	.	.	. ; UPPER CASE CHARACTER INPUT - EVALUATE SEQUENCE *
9126	22AB	.	.	. ;*****
9127	22AB	CD	E5 1A	CALL CHKSFK ;SOFT KEY DEFINE MODE?
9128	22AE	CC	17 23	CZ SWAP ;NO - SET TO SOFT KEY DISPLA
9129	22B1	3A	D9 FF	LDA PARM3 ;COMPUTE DESIRED KEY DATA RO
9130	22B4	87	.	ADD A
9131	22B5	3C	.	INR A ;= 2*(KEY NUMBER) + 1
9132	22B6	32	C0 FF	STA CURROW
9133	22B9	21	A6 FF	LXI H,SFTKYS ;LOCATE THE START OF THE
9134	22BC	CD	FF 0B	CALL MLKSC1 ;DATA ROW
9135	22BF	3E	50 .	MVI A,MAXCOL+1
9136	22C1	CD	D6 20	CALL FNDLS0 ;LOCATE THE END OF THE DATA
9137	22C4	3E	51 .	MVI A,MAXCOL+2 ;ROW + 1
9138	22C6	90	.	SUB B
9139	22C7	32	D9 FF	STA PARM3 ;SAVE END COLUMN NUMBER
9140	22CA	3A	DB FF	LDA PARM1 ;TRY TO EXTEND LINE TO
9141	22CD	32	C1 FF	STA CURCOL ;END OF NEW DATA LINE
9142	22D0	CD	08 25	CALL DSPASC ;TRY TO ALLOCATE LINE NEEDED
9143	22D3	B7	.	ORA A ;COLUMN POSITION ALLOCATED?
9144	22D4	CA	08 23	JZ DFS250 ;NO - DON'T SET NEW VALUE
9145	22D7	2A	C3 FF	LHLD CURADR ;YES - GET ADDRESS OF
9146	22DA	CD	8F 0C	CALL NXTCH0 ;END OF NEW DATA LINE
9147	22DD	3A	C1 FF	LDA CURCOL ;GET NUMBER OF DATA CHARS
9148	22E0	FE	4F .	CPI MAXCOL ;FULL LINE USED?
9149	22E2	C4	AD 1D	CNZ CLERLA ;NO - CLEAR EXCESS CHARACTER
9150	22E5	CD	7C 23	CALL CURPRT ;SET CURRENT COLUMN TO ZERO
9151	22E8	21	C0 FF	LXI H,CURROW ;SET FOR ATTRIBUTE ROW
9152	22EB	35	.	DCR M
9153	22EC	3A	DA FF	LDA PARM2 ;GET ATTRIBUTE PARAMETER
9154	22EF	3D	.	DCR A ;WHICH ATTRIBUTE TO SET?
9155	22F0	3E	4E .	MVI A,N ;(N = NORMAL)
9156	22F2	FA	FC 22	JM DFS230 ;0 - SET AS NORMAL KEY
9157	22F5	3E	4C .	MVI A,L ;(L = LOCAL ONLY)
9158	22F7	CA	FC 22	JZ DFS230 ;1 - SET FOR LOCAL ONLY
9159	22FA	3E	54 .	MVI A,T ;2 - SET FOR TRANSMIT ONLY
9160	22FC	.	.	. DFS230 EQU \$
9161	22FC	CD	7D 25	CALL DSPTST ;STORE ATTRIBUTE LETTER
9162	22FF	CD	11 23	CALL SWAP0 ;RESTORE ACTIVE DISPLAY
9163	2302	21	AB 7F	LXI H,DFSTB2 ;SET RANGE TABLE FOR SOFT
9164	2305	C3	32 05	JMP ESCAPA ;KEY DATA ACCUMULATION

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 282
9166	2308	. . .	;*****	
9167	2308	. . .	; NOT ENOUGH BLOCKS AVAILABLE FOR SUFT KEY DATA *	
9168	2308	. . .	; RESTORE OLD STATE AND IGNORE DEFINITION *	
9169	2308	. . .	;*****	
9170	2308	. . .	DFS250 EQU \$	
9171	2308	3A D9 FF	LDA PARM3 ;RECALL END OF DATA LINE	
9172	2308	32 C1 FF	STA CURCOL	
9173	230E	CD 95 1D	CALL CLEARL ;CLEAR ANY ADDED CHARACTERS	
9174	2311	. . .	;*****	
9175	2311	. . .	; SWAP - SWAP DISPLAY PARAMETERS BETWEEN SOFT *	
9176	2311	. . .	; KEY AND NORMAL DISPLAY *	
9177	2311	. . .	;*****	
9178	2311	. . .	;	
9179	2311	. . .	; ENTRY: DON'T CARE	
9180	2311	. . .	;	
9181	2311	. . .	; EXIT : DISPLAY PARAMTERS EXCHANGED	
9182	2311	. . .	; ALL REGISTERS DESTROYED	
9183	2311	. . .	;	
9184	2311	. . .	SWAPO EQU \$	
9185	2311	3A F8 FF	LDA CMFLGS ;GET COMMON FLAGS	
9186	2314	E6 08 .	ANI DEFSKY ;DEFINE SOFT KEY MODE?	
9187	2316	C0 . .	RNZ ;NO - DON'T DO SWAP	
9188	2317	. . .	;	
9189	2317	. . .	SWAP EQU \$	
9190	2317	21 AE FF	LXI H,DSPTYP ;SET DISPLAY TYPE FLAG	
9191	231A	7E . .	MOV A,M ;TO VALUE FOR DISPLAY TO	
9192	231B	2F . .	CMA ;MADE ACTIVE	
9193	231C	77 . .	MOV M,A	
9194	231D	. . .	SWAP1 EQU \$	
9195	231D	0E 0F .	MVI C,NUMSWP ;SET SWAP COUNT	
9196	231F	11 AF FF	LXI D,SWPSTR ;SET ADDRESS OF LOCATIONS	
9197	2322	21 BE FF	LXI H,RHTMGN ;TO BE EXCHANGED	
9198	2325	. . .	;	
9199	2325	. . .	; EXCHANGE DISPLAY PARAMETERS	
9200	2325	. . .	;	
9201	2325	. . .	SWP010 EQU \$	
9202	2325	46 . .	MOV B,M ;GET CURRENT SETTING	
9203	2326	1A . .	LDAX D ;GET STORED SETTING	
9204	2327	EB . .	XCHG ;EXCHANGE ADDRESSES	
9205	2328	70 . .	MOV M,B ;STORE NEW SAVE VALUE	
9206	2329	12 . .	STAX D ;STORE NEW CURRENT VALUE	
9207	232A	EB . .	XCHG ;RESTORE ADDRESSES	
9208	232B	13 . .	INX D ;INCREMENT TO NEXT VALUE	
9209	232C	23 . .	INX H	
9210	232D	0D . .	DCR C ;ALL VALUES EXCHANGED?	
9211	232E	C2 25 23	JNZ SWP010 ;NO - MOVE NEXT VALUE	
9212	2331	C9 . .	RET ;YES - RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
9214	2332	.	.	;*****
9215	2332	.	.	; SET SOFT KEY DATA *
9216	2332	.	.	;*****
9217	2332	.	.	DFS300 EQU \$
9218	2332	CD	E5 1A	CALL CHKSFK ;SOFT KEY ALREADY ENABLED?
9219	2335	CC	17 23	CZ SWAP ;NO - SET SOFT KEY DISPLAY 0
9220	2338	21	F4 FF	LXI H,MDFLG1 ;GET SOFT MODE FLAGS
9221	233B	7E	.	MOV A,M
9222	233C	F5	.	PUSH PSW ;SAVE SOFT MODE FLAGS
9223	233D	36	00 .	MVI M,0 ;FORCE INSERT CHARACTER OFF
9224	233F	CD	FA 15	CALL FDESC1 ;ADD INPUT TO DEFINITION
9225	2342	F1	.	POP PSW ;RECALL SOFT MODE FLAGS
9226	2343	32	F4 FF	STA MDFLG1 ;RESTORE ORIGINAL VALUES
9227	2346	CD	11 23	CALL SWAPO ;RESTORE ACTIVE DISPLAY
9228	2349	CD	10 06	CALL GETDC1 ;SET DISPLAY CURSOR
9229	234C	21	DB FF	LXI H,NEWCOL
9230	234F	35	.	DCR M ;ALL CHARACTERS DONE?
9231	2350	F2	42 05	JP ESCAP1 ;NO - CONTINUE ESC SEQUENCE
9232	2353	21	EF 7D	LXI H,DFSTB3 ;YES - SET TO WAIT FOR ANY
9233	2356	C3	32 05	JMP ESCAPA ;CHAR EXCEPT CR, LF, OR DC
9234	2359	.	.	;*****
9235	2359	.	.	; WAIT FOR CHARACTER TO RESTORE NORMAL MODE *
9236	2359	.	.	;*****
9237	2359	.	.	DFS350 EQU \$;LINE FEED CODE
9238	2359	CD	D7 13	CALL DCXB2D ;DATA FROM KEYBOARD?
9239	235C	CA	6F 0B	JZ LNFEED ;YES - DO LINE FEED
9240	235F	C9	.	RET ;NO - RETURN TO RE-ENABLE AL
9241	2360	.	.	; CODES BY CALL TO "ESCEND"
9242	2360	.	.	; IN "CHINT" CLEAN-UP
9243	2360	.	.	;
9244	2360	.	.	DFS360 EQU \$;RETURN CODE
9245	2360	CD	D7 13	CALL DCXB2D ;DATA FROM KEYBOARD
9246	2363	C2	42 05	JNZ ESCAP1 ;NO - CONTINUE WAITING
9247	2366	.	.	; YES - DO RETURN OPERATION
9248	2366	.	.	;*****
9249	2366	.	.	; CRRET - SET CURSOR TO LEFT MARGIN *
9250	2366	.	.	;*****
9251	2366	.	.	;
9252	2366	.	.	; ENTRY: DON'T CARE
9253	2366	.	.	;
9254	2366	.	.	; EXIT : A,CURCOL = LEFT MARGIN SETTING
9255	2366	.	.	; IF SPOW NOT DISABLED, SPOW SET
9256	2366	.	.	;
9257	2366	.	.	CRRET EQU \$
9258	2366	.	.	;***** GRAPHICS MODIFICATION *****
9259	2366	3A	97 90	LDA ZGFLG6 ;IN GRAPHICS TEXT MODE?
9260	2369	E6	02 .	ANI GTEXT
9261	236B	C4	20 60	CNZ ZCR ;PROCESS IN A/N?
9262	236E	D8	.	RC ;NO, PROCESS IN GRAPHICS
9263	236F	.	.	;*****

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 284
=====
9264     236F     3A  FB  FF          LDA  KBJMPR      ;GET STRAP SETTINGS
9265     2372     E6  02  .          ANI  SPLDIS      ;SPOW DISABLED?
9266     2374     CA  7C  23         JZ   CURPRT      ;YES - RETURN CURSOR ONLY
9267     2377     21  6C  FF         LXI  H,SPOWL    ;NO - SET SPOW LATCH
9268     237A     36  20  .          MVI  M,SPOWON
9269     237C     .   .   .          CURPRT EQU $
9270     237C     3A  BF  FF          LDA  LFTMGN     ;SET CURSOR TO LEFT MARGIN
9271     237F     .   .   .          CRRET1 EQU $
9272     237F     32  C1  FF          STA  CURCOL     ;UPDATE CURRENT COLUMN NUMBE
9273     2382     32  00  87          STA  IOCRCL     ;AND SET DISPLAY CURSOR
9274     2385     C9  .   .          RET             ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 285
9276	2386	.	.	*****	
9277	2386	.	.	; EXSFKY--EXECUTE SOFT KEY. CANNOT DEFINE AND	
9278	2386	.	.	; EXECUTE KEY IN SAME SEQUENCE	
9279	2386	.	.	; SEQUENCE IS ESC & F <KEY#> E	
9280	2386	.	.	*****	
9281	2386	.	.	EXSFKY EQU \$	
9282	2386	CD	E5 1A	CALL CHKSFK ;SOFT KEYS UP?	
9283	2389	CO	.	RNZ ;YES, DONT EXECUTE	
9284	238A	.	.	; IF SOFT KEY IS ALSO BEING DEFINED, ABORT	
9285	238A	.	.	; SEQUENCE	
9286	238A	3A	76 FE	LDA SKFLGS ;DEFINITION IN PROGRESS?	
9287	238D	4F	.	MOV C,A ;(SAVE FLAGS IN C)	
9288	238E	E6	01 .	ANI DEFKEY	
9289	2390	C2	48 05	JNZ ESCEND ;YES, ABORT SEQUENCE	
9290	2393	.	.	; GET KEY NUMBER	
9291	2393	2A	DE FF	LHLD IODATA	
9292	2396	7C	.	MOV A,H ;MSBYTE SHOULD BE 0	
9293	2397	B7	.	ORA A	
9294	2398	C2	48 05	JNZ ESCEND ;TOO BIG, ABORT	
9295	2398	7D	.	MOV A,L ;CHECK KEY #	
9296	239C	FE	09 .	CPI NMFCTK ;IN RANGE?	
9297	239E	D2	48 05	JNC ESCEND ;NO, ABORT	
9298	23A1	C6	EF .	ADI FOCODE ;CONVERT TO KEY CODE	
9299	23A3	.	.	; A = KEY # OF SOFT KEY TO BE EXECUTED	
9300	23A3	32	75 FE	STA SFTKEY ;STORE CURRENT SOFT KEY	
9301	23A6	.	.	; TEST FOR SOFT KEY CURRENTLY BEING EXECUTED	
9302	23A6	3E	02 .	MVI A,SKIP ;SOFT KEY ALREADY IN	
9303	23A8	A1	.	ANA C ;PROGRESS?	
9304	23A9	C2	EA 23	JNZ EXK050 ;YES, PROCESS NEW KEY	
9305	23AC	3E	02 .	MVI A,SKIP ;NO, SET KEY IN PROGRESS	
9306	23AE	CD	FD 23	CALL STSKFL ;FLAG	
9307	23B1	.	.	; EXECUTE THE NEW SOFT KEY	
9308	23B1	.	.	EXK010 EQU \$	
9309	23B1	CD	48 05	CALL ESCEND ;RESET RANGE TABLES	
9310	23B4	3A	75 FE	LDA SFTKEY ;GET SOFT KEY KEY CODE	
9311	23B7	4F	.	MOV C,A	
9312	23B8	.	.	;	
9313	23B8	.	.	EXSK1 EQU \$;ENTRY FROM WAITLOOP	
9314	23B8	.	.	;	
9315	23B8	.	.	; IF KEY IS L OR T, FCTKEY WILL PROCESS THE ENTIRE	
9316	23B8	.	.	; KEY. IF N, IT MUST BE PROCESSED CHAR BY CHAR.	
9317	23B8	.	.	; EXECUTION OF KEY MAY TRIGGER ANOTHER ONE	
9318	23B8	CD	4A 15	CALL FCTKEY ;PROCESS IT	
9319	23BB	3A	76 FE	LDA SKFLGS ;WAS A NEW KEY TRIGGERED?	
9320	23BE	E6	04 .	ANI NEWKEY	
9321	23C0	CA	C9 23	JZ EXK020 ;NO, CHECK FOR N KEY	
9322	23C3	.	.	; NEW KEY TRIGGERED, START EXECUTING IT	
9323	23C3	.	.	EXK015 EQU \$	
9324	23C3	CD	03 24	CALL CLSKFL ;CLEAR NEW KEY FLAG	
9325	23C6	C3	B1 23	JMP EXK010 ;EXECUTE NEW KEY	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 286
9326	23C9	. . .	EXK020 EQU \$	
9327	23C9	. . .	; CHECK FOR N KEY. IF NOT, SOFT KEY HAS BEEN COMP	
9328	23C9	. . .	; LETELY PROCESSED	
9329	23C9	3A 6E FF	LDA DFLGS	
9330	23CC	E6 10 .	ANI FCTK2D ;N KEY?	
9331	23CE	CA E5 23	JZ EXK040 ;NO, KEY IS DONE	
9332	23D1	. . .	; PROCESS N KEY CHAR BY CHAR	
9333	23D1	. . .	EXK030 EQU \$	
9334	23D1	CD 94 16	CALL GTFCTK ;GET THE NEXT CHAR	
9335	23D4	CA E5 23	JZ EXK040 ;NONE LEFT, DONE	
9336	23D7	CD 0A 24	CALL LOCLN2 ;PROCESS CHAR	
9337	23DA	3A 76 FE	LDA SKFLGS ;WAS A NEW SOFT KEY	
9338	23DD	E6 04 .	ANI NEWKEY ;TRIGGERED?	
9339	23DF	CA D1 23	JZ EXK030 ;NO, PROCESS NEXT CHAR	
9340	23E2	C3 C3 23	JMP EXK015 ;YES, START NEW KEY	
9341	23E5	. . .	; SOFT KEY DONE	
9342	23E5	. . .	EXK040 EQU \$	
9343	23E5	3E 02 .	MVI A,SKIP ;CLEAR SOFT KEY IN PROGRESS	
9344	23E7	C3 03 24	JMP CLSKFL ;FLAG	
9345	23EA	. . .	; SOFT KEY TRIGGERED DURING EXECUTION OF ANOTHER	
9346	23EA	. . .	; FLAG THE NEW KEY, AND TERMINATE EXECUTION OF	
9347	23EA	. . .	; THE PREVIOUS ONE	
9348	23EA	. . .	EXK050 EQU \$	
9349	23EA	3E 04 .	MVI A,NEWKEY ;SET NEW KEY FLAG	
9350	23EC	CD FD 23	CALL STSKFL	
9351	23EF	. . .	; TERMINATE CURRENT KEY	
9352	23EF	. . .	EXK060 EQU \$	
9353	23EF	CD 94 16	CALL GTFCTK ;GO THRU SOFT KEY	
9354	23F2	C2 EF 23	JNZ EXK060 ;UNTIL NONE LEFT	
9355	23F5	2A A4 FF	LHLD CURFKY ;NEXT CALL TO	
9356	23F8	23 . .	INX H ;GTFCTK WILL RETURN END	
9357	23F9	22 A4 FF	SHLD CURFKY ;OF SOFT KEY	
9358	23FC	C9 . .	RET	
9359	23FD	. . .	;*****	
9360	23FD	. . .	; STSKFL--SET BIT IN SOFT KEY FLAGS	
9361	23FD	. . .	; ENTRY A = BIT(S) TO BE SET	
9362	23FD	. . .	;*****	
9363	23FD	. . .	STSKFL EQU \$	
9364	23FD	21 76 FE	LXI H,SKFLGS	
9365	2400	B6 . .	ORA M	
9366	2401	77 . .	MOV M,A	
9367	2402	C9 . .	RET	
9368	2403	. . .	;*****	
9369	2403	. . .	; CLSKFL--CLEAR BIT IN SOFT KEY FLAGS	
9370	2403	. . .	; ENTRY A = BIT(S) TO BE CLEARED	
9371	2403	. . .	;*****	
9372	2403	. . .	CLSKFL EQU \$	
9373	2403	21 76 FE	LXI H,SKFLGS	
9374	2406	2F . .	CMA	
9375	2407	A6 . .	ANA M	

13255
2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
9376     2408     77 . .      MOV  M,A
9377     2409     C9 . .      RET
=====
```

PAGE 287

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 288
=====
9379      240A      . . .      ;*****
9380      240A      . . .      ; LOCLN2--PROCESS CHAR THRU LOCLIN, BUT ALSO DO
9381      240A      . . .      ; TEST FOR AUTO LINE FEED, AND MONITOR TAPES
9382      240A      . . .      ; AND DATA COM
9383      240A      . . .      ; ENTRY A = CHAR
9384      240A      . . .      ;*****
9385      240A      . . .      LOCLN2 EQU $
9386      240A      32 9C FF      STA CHARIN      ;STORE INPUT CHAR
9387      2400      4F . .      MOV C,A
9388      240E      CD CF 06      CALL LOCLIN     ;PROCESS IT
9389      2411      3A 9C FF      LDA CHARIN     ;TEST FOR AUTO LINE FEED
9390      2414      FE 0D .      CPI CR         ;WAS CHAR RETURN?
9391      2416      C2 2B 24      JNZ LCN010     ;NO
9392      2419      3A F3 FF      LDA MDFLG2     ;YES, IS AUTO LINEFEED ON?
9393      241C      E6 04 .      ANI AUTOLF
9394      241E      CA 2B 24      JZ LCN010      ;NO, DONT DO LINE FEED
9395      2421      2E 01 .      MVI L,1        ;YES, FIRST WAIT 10 MS
9396      2423      CD E4 13      CALL DELAY
9397      2426      3E 0A .      MVI A,LF       ;THEN FAKE A LINE FEED
9398      2428      C3 0A 24      JMP LOCLN2
9399      2428      . . .      LCN010 EQU $
9400      2428      CD 88 05      CALL GETDCM    ;MONITOR DATACOM
9401      242E      CD 08 16      CALL IUCTMN    ;MONITOR TAPES
9402      2431      C9 . .      RET
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
9404	2432	.	.	*****
9405	2432	.	.	; DISPLAY ENHANCEMENT *
9406	2432	.	.	*****
9407	2432	.	.	DISPEN EQU \$
9408	2432	CD	E5 1A	CALL CHKSFK ;DEFINE SOFT KEY MODE?
9409	2435	C0	.	RNZ ;YES - NO DISPLAY ENHANCEMEN
9410	2436	21	0B 7F	LXI H,DENTAB ;SET FOR DISPLAY ENHANCEMENT
9411	2439	C3	34 05	JMP ESCAPO
9412	243C	.	.	*****
9413	243C	.	.	; DISPLC - ENTER DISPLAY ENHANCEMENT CHAR *
9414	243C	.	.	*****
9415	243C	.	.	DISPLC EQU \$
9416	243C	3A	89 FF	LDA DCHAR ;GET DISPLAY CHARACTER
9417	243F	E6	0F .	ANI 170 ;EXTRACT ENHANCEMENT BITS
9418	2441	.	.	DISPC0 EQU \$
9419	2441	06	30 .	MVI B,600 ;SET MASK TO SAVE ALT CHAR
9420	2443	.	.	*****
9421	2443	.	.	; DISPC1 - ENTER ENHANCEMENT OR FLAG CHARACTER *
9422	2443	.	.	*****
9423	2443	.	.	;
9424	2443	.	.	; ENTRY: A = CHARACTER TO BE STORED
9425	2443	.	.	; B = MASK TO SAVE UNCHANGED PART (USED
9426	2443	.	.	; ONLY FOR ENHANCEMENT CHARACTERS)
9427	2443	.	.	;
9428	2443	.	.	; EXIT : SEE "DISPLA"
9429	2443	.	.	;
9430	2443	.	.	DISPC1 EQU \$
9431	2443	F6	80 .	ORI 2000 ;ADD BIT FOR REFRESH LOGIC
9432	2445	.	.	DISPC2 EQU \$
9433	2445	32	89 FF	STA DCHAR ;STORE NEW ENHANCEMENT CODE
9434	2448	78	.	MOV A,B ;STORE MASK FOR ENHANCEMENT
9435	2449	32	77 FF	STA CDSPEN ;BITS NOT TO BE ALTERED
9436	244C	.	.	; FALL INTO DISPLAY ROUTINE

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 290
9438	244C	.	.	. ;*****	
9439	244C	.	.	. ; DISPLA - ADD CHARACTER TO DISPLAY *	
9440	244C	.	.	. ;*****	
9441	244C	.	.	. ;	
9442	244C	.	.	. ; ENTRY: CURCOL,CURROW = SCREEN POSITION WHERE	
9443	244C	.	.	. ; CHARACTER IS TO BE INSERTED	
9444	244C	.	.	. ; DCHAR = CHARACTER TO BE DISPLAYED	
9445	244C	.	.	. ; CDSPEN = MASK TO MASK OUT COMMON BITS	
9446	244C	.	.	. ; IF DCHAR IS A DISPLAY CONTROL BYTE	
9447	244C	.	.	. ;	
9448	244C	.	.	. ; EXIT : A = 0, NO PLACE FOR CHARACTER	
9449	244C	.	.	. ; A # 0, CHARACTER PROCESSED	
9450	244C	.	.	. ; B = CHARACTER REPLACED IF ADDITION	
9451	244C	.	.	. ; DONE BY INSERT	
9452	244C	.	.	. ; D,E = ADDRESS OF CHAR IN DISPLAY	
9453	244C	.	.	. ;	
9454	244C	.	.	. DISPLA EQU \$	
9455	244C	3A	89	FF LDA DCHAR ;GET CHAR TO BE STORED	
9456	244F	B7	.	. ORA A ;IS THIS ASCII CHAR?	
9457	2450	F2	08	25 JP DIS060 ;YES - CONTINUE	
9458	2453	.	.	. ;*****	
9459	2453	.	.	. ; CONTROL CODE TO BE ENTERED INTO *	
9460	2453	.	.	. ; DATA STREAM - FIND CHAR PRECEDING *	
9461	2453	.	.	. ; THIS COLUMN *	
9462	2453	.	.	. ;*****	
9463	2453	3A	C1	FF LDA CURCOL ;GET CURRENT COLUMN NUMBER	
9464	2456	3D	.	. DCR A ;SET FOR PREVIOUS COLUMN	
9465	2457	CD	10	08 CALL RCADRO ;DOES LINE EXIST?	
9466	245A	FA	1F	0C JM MLOCK1 ;NO - SOUND BELL AND EXIT	
9467	245D	.	.	. ; WITH A-REGISTER = 0	
9468	245D	C2	55	25 JNZ DIS100 ;COL BEYOND EOL - EXTEND LIN	
9469	2460	.	.	. ;*****	
9470	2460	.	.	. ; PREVIOUS COLUMN FOUND *	
9471	2460	.	.	. ;*****	
9472	2460	4F	.	. MOV C,A ;SAVE COLUMN IN C	
9473	2461	0C	.	. INR C ;SET C TO NEXT COLUMN NUMBER	
9474	2462	CD	90	11 CALL CKPROT ;PREVIOUS CHAR PROTECTED?	
9475	2465	C2	76	24 JNZ DIS030 ;NO - CONTINUE	
9476	2468	.	.	. DIS020 EQU \$	
9477	2468	1B	.	. DCX D ;YES - SET PTR TO NEXT CHAR	
9478	2469	21	C2	C1 LXI H,ENDPR*256+XMONLY	
9479	246C	CD	1E	20 CALL FNDCH ;IS NEXT CHARACTER PROTECTED	
9480	246F	CA	3D	25 JZ DIS092 ;YES - LOOK FOR NEXT FIELD	
9481	2472	21	C1	FF LXI H,CURCOL ;YES - RECALL COLUMN VALUE	
9482	2475	4E	.	. MOV C,M	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
9484	2476	.	.	*****
9485	2476	.	.	; SEARCH FOR PLACE FOR CHARACTER *
9486	2476	.	.	*****
9487	2476	.	.	DIS030 EQU \$
9488	2476	CD	90 0C	CALL NXTCHR ;GET NEXT CHAR
9489	2479	47	.	MOV B,A ;SAVE EXISTING CHAR IN B-REG
9490	247A	21	89 FF	LXI H,DCHAR
9491	247D	FE	C4 .	CPI STPFLG ;NON-DISPLAYING TERMINATOR?
9492	247F	CA	A2 24	JZ DIS035 ;YES - DELETE IT
9493	2482	FE	CC .	CPI EOL ;EXISTING CHARACTER AN EOL?
9494	2484	7E	.	MOV A,M ;(GET CHAR TO BE DISPLAYED
9495	2485	CA	EA 24	JZ DIS050 ;YES - ADD CHARACTER TO LINE
9496	2488	FE	C4 .	CPI STPFLG ;NON-DISPLAYING TERMINATOR?
9497	248A	CA	2E 1C	JZ CRI104 ;YES - INSERT TERMINATOR
9498	248D	78	.	MOV A,B ;NO - RECALL EXISTING CHAR
9499	248E	87	.	ADD A ;EXISTING CHARACTER ASCII?
9500	248F	7E	.	MOV A,M ;(GET CHAR TO BE DISPLAYED
9501	2490	D2	EA 24	JNC DIS050 ;YES - INSERT NEW CHARACTER
9502	2493	FA	AA 24	JM DIS040 ;FLAG CHAR - ADD FLAG TO DIS
9503	2496	87	.	ADD A ;NEW CHAR DISPLAY CONTROL?
9504	2497	FA	76 24	JM DIS030 ;NO - GO TO NEXT CHARACTER
9505	249A	.	.	*****
9506	249A	.	.	; MERGE NEW DISPLAY ENHANCEMENT *
9507	249A	.	.	; WITH CODE ALREADY IN THIS COLUMN *
9508	249A	.	.	*****
9509	249A	3A	77 FF	LDA CDSPEN ;GET ENHANCEMENT MASK
9510	249D	A0	.	ANA B ;EXTRACT BITS TO BE SAVED
9511	249E	B6	.	ORA M ;COMBINE WITH NEW ENHANCEMEN
9512	249F	C3	D8 24	JMP DIS044 ;STORE THE NEW DISPLAY CODE
9513	24A2	.	.	*****
9514	24A2	.	.	; NON-DISPLAYING TERMINATOR FOUND - DELETE IT *
9515	24A2	.	.	*****
9516	24A2	.	.	DIS035 EQU \$
9517	24A2	BE	.	CMP M ;IS NEW CHAR TERMINATOR ALSO
9518	24A3	C8	.	RZ ;YES - RETURN
9519	24A4	CD	11 1C	CALL CHRDL2 ;NO - DELETE THE CHARACTER
9520	24A7	C3	76 24	JMP DIS030 ;CONTINUE SCAN

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 292
=====
9522     24AA      . . .      ;*****
9523     24AA      . . .      ; FLAG CHAR FOUND *
9524     24AA      . . .      ;*****
9525     24AA      . . .      DIS040 EQU $
9526     24AA      B8 . . .    CMP B      ;IS THIS SAME FLAG CHAR?
9527     24AB      C8 . . .    RZ         ;YES - RETURN (A # 0)
9528     24AC      87 . . .    ADD A      ;NEW CHARACTER DISPLAY CNTL?
9529     24AD      1F . . .    RAR        ;(RESTORE CHARACTER)
9530     24AE      F2 DB 24    JP DIS045  ;YES - CHECK PROTECTED FIELD
9531     24B1      FE C5 .    CPI ALPHA  ;IS NEW CHAR TYPE DEFINITION
9532     24B3      78 . . .    MOV A,B    ;(RECALL OLD FLAG CHAR)
9533     24B4      FA BF 24    JM DIS042  ;NO - ADD FIELD DEFINITION
9534     24B7      FE C5 .    CPI ALPHA  ;IS OLD CHAR TYPE DEFINITION
9535     24B9      F2 CE 24    JP DIS043  ;YES - REPLACE THE CHARACTER
9536     24BC      C3 76 24    JMP DIS030 ;NO - GO TO NEXT CHARACTER
9537     24BF      . . .      ;*****
9538     24BF      . . .      ; FIELD DEFINITION CHARACTER TO BE ADDED - PUT *
9539     24BF      . . .      ; AHEAD OF TYPE DEFINITION OR AFTER "STPR" *
9540     24BF      . . .      ;*****
9541     24BF      . . .      DIS042 EQU $
9542     24BF      FE C5 .    CPI ALPHA  ;IS OLD CHAR TYPE DEFINITION
9543     24C1      F2 2E 1C   JP CRI104  ;YES - INSERT FIELD DEF
9544     24C4      3E C0 .    MVI A,STPR ;NO - STORE NEW FIELD DEF
9545     24C6      B8 . . .    CMP B      ;OLD CHAR = START PROTECT?
9546     24C7      CA 76 24   JZ DIS030  ;YES - LOOK TO NEXT CHAR
9547     24CA      BE . . .    CMP M      ;IS NEW CHAR A STPR?
9548     24CB      CA 2E 1C   JZ CRI104  ;YES - INSERT BEFORE UNPROTC
9549     24CE      . . .      ;*****
9550     24CE      . . .      ; REPLACE EXISTING DISPLAY CHARACTER *
9551     24CE      . . .      ;*****
9552     24CE      . . .      DIS043 EQU $
9553     24CE      1A . . .    LDAX D     ;PUT EXISTING CHARACTER INTO
9554     24CF      47 . . .    MOV B,A    ;B-REG FOR SOFT KEY CHECK
9555     24D0      3A 89 FF   LDA DCHAR  ;GET CHAR TO BE DISPLAYED
9556     24D3      21 6C FF   LXI H,SPOWL ;CHECK AGAINST SPOW LATCH
9557     24D6      BE . . .    CMP M      ;INPUT = SPACE AND SPOW SET?
9558     24D7      C8 . . .    RZ         ;YES - RETURN (A # 0)
9559     24D8      . . .      DIS044 EQU $
9560     24D8      12 . . .    STAX D     ;STORE THE NEW CHARACTER
9561     24D9      3C . . .    INR A      ;FORCE A # 0
9562     24DA      C9 . . .    RET        ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 293
9564	24DB	.	.	*****	
9565	24DB	.	.	; FLAG CHAR FOUND AND *	
9566	24DB	.	.	; DISPLAY CONTROL TO BE ADDED *	
9567	24DB	.	.	*****	
9568	24DB	.	.	DIS045 EQU \$	
9569	24DB	78	.	MOV A,B ;RECALL EXISTING CHARACTER	
9570	24DC	FE	C0	CPI STPR ;BEGINNING A PROTECTED FIELD	
9571	24DE	C2	76 24	JNZ DIS030 ;NO - MOVE TO NEXT CHAR	
9572	24E1	CD	CF 1A	CALL CHKFMS ;FORMAT MODE?	
9573	24E4	CA	76 24	JZ DIS030 ;NO - ADD CHAR TO DISPLAY	
9574	24E7	C3	68 24	JMP DIS020 ;YES - LOOK FOR NEXT FIELD	
9575	24EA	.	.	*****	
9576	24EA	.	.	; ASCII OR EOL FOUND *	
9577	24EA	.	.	; MERGE NEW DISPLAY CONTROL IF NECESSARY *	
9578	24EA	.	.	*****	
9579	24EA	.	.	DIS050 EQU \$	
9580	24EA	87	.	ADD A ;NEW CHAR DISPLAY CONTROL?	
9581	24EB	FA	F8 24	JM DIS054 ;NO - ADD CHAR TO DISPLAY	
9582	24EE	3A	77 FF	LDA CDSPEN ;YES - GET MASK	
9583	24F1	2E	C6	MVI L,LSTDCD-BASE ;GET LAST ENHANCEMENT	
9584	24F3	A6	.	ANA M ;EXTRACT BITS TO BE SAVED	
9585	24F4	2E	89	MVI L,DCHAR-BASE	
9586	24F6	B6	.	ORA M ;COMBINE WITH NEW ENHANCEMEN	
9587	24F7	77	.	MOV M,A ;STORE	
9588	24F8	.	.	DIS054 EQU \$	
9589	24F8	78	.	MOV A,B ;WAS CHAR ASCII?	
9590	24F9	B7	.	ORA A	
9591	24FA	.	.	*****	
9592	24FA	.	.	; AUTO PLOT KLUGE TO KEEP TRACK OF HOW MANY	
9593	24FA	.	.	; CHARACTERS ARE INSERTED	
9594	24FA	FA	03 25	JM DIS057 ;NO, ADD SINGLE CHAR	
9595	24FD	CD	26 60	CALL ZINFIX	
9596	2500	C3	2E 1C	JMP CRI104 ;YES, DO INSERT	
9597	2503	.	.	*****	
9598	2503	.	.	DIS057 EQU \$	
9599	2503	0E	00	MVI C,0 ;NO - ADD SINGLE CHAR	
9600	2505	C3	6A 25	JMP DIS110	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 294
=====
9602     2508      . . .      ;*****
9603     2508      . . .      ; ENTER ASCII CHARACTER INTO DATA STREAM *
9604     2508      . . .      ;*****
9605     2508      . . .      DSPASC EQU $
9606     2508      . . .      DIS060 EQU $
9607     2508      CD 0D 08      CALL RCADDR ;GET MEMORY ADDRESS
9608     2508      CA 27 25      JZ DIS080 ;CHAR FOUND BY RCADDR
9609     250E      . . .      DISPL0 EQU $
9610     250E      FA 1D 25      JM DIS070 ;RETURN IF LINE NOT BUILT
9611     2511      OD . .      DCR C
9612     2512      C2 55 25      JNZ DIS100 ;MORE THAN ONE CHAR NEEDED
9613     2515      . . .      ;*****
9614     2515      . . .      ; SINGLE CHARACTER REQUIRED *
9615     2515      . . .      ; CHECK FOR LAST COLUMN OF LINE *
9616     2515      . . .      ;*****
9617     2515      FE 4F .      CPI MAXCOL ;COMPARE WITH MAX COLUMN
9618     2517      C2 6A 25      JNZ DIS110 ;NOT MAXIMUM COLUMN
9619     251A      C3 37 25      JMP DIS090
9620     251D      . . .      ;*****
9621     251D      . . .      ; LINE NOT BUILT *
9622     251D      . . .      ; PERFORM HOMEUP IF FORMAT MODE *
9623     251D      . . .      ;*****
9624     251D      . . .      DIS070 EQU $
9625     251D      CD CF 1A      CALL CHKFMS ;FORMAT MODE?
9626     2520      C8 . .      RZ ;NO - RETURN (A = 0)
9627     2521      CD 14 48      CALL ZBELL ;YES - SOUND BELL
9628     2524      C3 4E 25      JMP DIS093 ;HOME UP AND TRY AGAIN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS
9630	2527	.	.	.	*****
9631	2527	.	.	.	; CHARACTER REPLACEMENT *
9632	2527	.	.	.	*****
9633	2527	.	.	.	DIS080 EQU \$
9634	2527	4F	.	.	MOV C,A ;SAVE COLUMN IN C
9635	2528	3A	F4	FF	LDA MDFLG1 ;GET TERMINAL MODE FLAGS
9636	2528	E6	02	.	ANI INSCHR ;IN CHARACTER INSERT MODE?
9637	252D	CA	37	25	JZ DIS090 ;NO - ADD CHARACTER TO DISPL
9638	2530	3A	90	FF	LDA EOLMV ;YES - GET EOL SHIFTED FLAG
9639	2533	B7	.	.	ORA A ;HAS LINE BEEN EXTENDED?
9640	2534	CA	28	1C	JZ CRI100 ;NO - PERFORM INSERT CHAR
9641	2537	.	.	.	DIS090 EQU \$
9642	2537	CD	90	11	CALL CKPROT ;CURSOR IN PROTECTED FIELD?
9643	253A	C2	CE	24	JNZ DIS043 ;NO - STORE THE CHARACTER
9644	253D	.	.	.	DIS092 EQU \$
9645	253D	CD	D7	13	CALL DCXB2D ;DATA COMM OR I/O BUFF CHAR?
9646	2540	CC	14	48	CZ ZBELL ;NO - SOUND THE BELL
9647	2543	3A	89	FF	LDA DCHAR ;GET CHAR TO BE DISPLAYED
9648	2546	B7	.	.	ORA A ;IS IT A CONTROL CHARACTER?
9649	2547	F8	.	.	RM ;YES - DON'T TAB (RETURN A#0)
9650	2548	CD	3A	1F	CALL FLDSR ;NO - TAB TO NEXT FIELD
9651	254B	C2	4C	24	JNZ DISPLA ;JUMP IF FIELD FOUND
9652	254E	.	.	.	DIS093 EQU \$
9653	254E	CD	A2	1E	CALL CURPH1 ;ANY FIELDS IN DISPLAY?
9654	2551	C2	4C	24	JNZ DISPLA ;YES - ADD CHARACTER TO FIEL
9655	2554	C9	.	.	RET ;NO - RETURN (A # 0)

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
9657      2555      . . .      ;*****
9658      2555      . . .      ; LINE MUST BE EXTENDED TO ACCOMODATE CHARACTER *
9659      2555      . . .      ;   - EXTEND TO ONE COLUMN BEFORE DESIRED COLUMN *
9660      2555      . . .      ;*****
9661      2555      . . .      ;
9662      2555      . . .      ; ENTRY:  C = NUMBER OF CHARACTERS REQUIRED
9663      2555      . . .      ;
9664      2555      . . .      DIS100 EQU $
9665      2555      0D . .      DCR C          ;MORE THAN ONE CHAR TO ADD?
9666      2556      C2 6A 25      JNZ DIS110     ;NO - ADD MULTIPLE CHARACTER
9667      2559      CD 90 11      CALL CKPROT    ;CURSOR IN PROTECTED FIELD?
9668      255C      CA 3D 25      JZ DIS092     ;YES - TAB TO NEXT FIELD
9669      255F      21 9B FF      LXI H,NCHAR   ;NO - SET "NCHAR" TO STORE
9670      2562      36 01 .      MVI M,1       ;BLANK OVER EOL (I.E.,
9671      2564      . . .      ;          MAKE DISPLAY ROUTINE
9672      2564      . . .      ;          THINK MORE THAN ONE
9673      2564      . . .      ;          CHARACTER BEING ADDED)
9674      2564      CD 87 09      CALL DISPL2   ;EXTEND LINE BY ONE CHARACTE
9675      2567      C3 73 25      JMP DIS114    ;CHECK MEMORY LOCKED
9676      256A      . . .      ;
9677      256A      . . .      DIS110 EQU $
9678      256A      CD 90 11      CALL CKPROT   ;CURSOR IN PROTECTED FIELD?
9679      256D      CA 3D 25      JZ DIS092    ;YES - TAB TO NEXT FIELD
9680      2570      CD AB 09      CALL DISPL1   ;NO - EXTEND LINE
9681      2573      . . .      DIS114 EQU $
9682      2573      B7 . .      ORA A        ;MEMORY LOCKED?
9683      2574      C8 . .      RZ          ;YES - RETURN FAIL (A = 0)
9684      2575      3A 9B FF      LDA NCHAR    ;GET # OF CHARACTERS ADDED
9685      2578      3D . .      DCR A       ;SINGLE CHARACTER ADDED?
9686      2579      F2 4C 24      JP DISPLA    ;NO - TRY TO STORE AGAIN
9687      257C      C9 . .      RET        ;YES - STORE DONE BY DISPLAY
9688      257D      . . .      ;          (A # 0)
=====

```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 298
=====
9721     2597     CD  CF  1A      CALL  CHKFMS      ;FORMAT/SOFT KEY DEFINE MODE
9722     259A     C8  .   .       RZ                ;NO - DO NORMAL EXIT
9723     259B     CD  D7  13      CALL  DCXB2D      ;DATA CUMM OR I/O BUFF CHAR?
9724     259E     C0  .   .       RNZ               ;NO - DO NORMAL EXIT
9725     259F     3A  89  FF      LDA  DCHAR        ;GET CHARACTER DISPLAYED
9726     25A2     2A  86  FF      LHLD CHKRTN       ;NO - GET CHECK ROUTINE ADDR
9727     25A5     CF  .   .       RST  RSTJMP       ;IS IT A VALID CHARACTER?
9728     25A6     C8  .   .       RZ                ;YES - DO NORMAL EXIT
9729     25A7     F1  .   .       POP  PSW          ;NO - POP OFF NORMAL EXIT AD
9730     25A8     .   .   .       ;
9731     25A8     .   .   .       ; FIELD CHECK ERROR - LOCK UP UNTIL BACKSPACE HIT
9732     25A8     .   .   .       ;
9733     25A8     AF  .   .       XRA  A            ;CLEAR OUT INPUT CHARACTER
9734     25A9     32  9C  FF      STA  CHARIN       ;TO KILL FUNCTION KEYS
9735     25AC     .   .   .       DCH010 EQU $
9736     25AC     CD  14  48      CALL  ZBELL        ;SOUND BELL
9737     25AF     .   .   .       DCH020 EQU $
9738     25AF     CD  D8  16      CALL  IOCTMN       ;MONITOR THE TAPE DRIVES
9739     25B2     CD  05  48      CALL  ZGETKY       ;ANY KEY HIT?
9740     25B5     C2  AF  25      JNZ  DCH020        ;NO - CONTINUE WAITING
9741     25B8     .   .   .       ;***** GRAPHICS MODIFICATION *****
9742     25B8     FE  EF  .       CPI  SFTCR         ;IS IT THE RETURN KEY?
9743     25BA     .   .   .       ;*****
9744     25BA     C2  AC  25      JNZ  DCH010        ;NO - SOUND BELL, TRY AGAIN
9745     25BD     3E  09  .       MVI  A,STPRPT     ;YES - STOP RETURN KEY
9746     25BF     C3  08  48      JMP  ZKCTL         ;FROM REPEATING AND EXIT
9747     25C2     .   .   .       ;*****
9748     25C2     .   .   .       ; CHARACTER NOT DISPLAYED - SOUND BELL IF *
9749     25C2     .   .   .       ; CHARACTER FROM KEYBOARD *
9750     25C2     .   .   .       ;*****
9751     25C2     .   .   .       DCH100 EQU $
9752     25C2     E1  .   .       POP  H            ;POP OFF NORMAL EXIT ROUTINE
9753     25C3     .   .   .       DSPCH1 EQU $
9754     25C3     CD  D7  13      CALL  DCXB2D      ;INPUT FROM KEYBOARD
9755     25C6     CA  14  48      JZ   ZBELL        ;YES - SOUND BELL AND EXIT
9756     25C9     C9  .   .       RET               ;NO - RETURN ONLY
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 299
9758	25CA	.	.	*****	
9759	25CA	.	.	; EXPAND - EXPAND DISPLAY CONTROL TO ESCAPE *	
9760	25CA	.	.	; SEQUENCE *	
9761	25CA	.	.	*****	
9762	25CA	.	.	;	
9763	25CA	.	.	; ENTRY: A,C = DISPLAY CONTROL BYTE	
9764	25CA	.	.	;	
9765	25CA	.	.	; EXIT : H = BASEH	
9766	25CA	.	.	; A,B,L DESTROYED	
9767	25CA	.	.	;	
9768	25CA	.	.	EXPAND EQU \$	
9769	25CA	CD	97 7D	CALL INITD1 ;INITIALIZE CHAR BUFFER PTRS	
9770	25CD	87	.	ADD A ;IS CHAR DISPLAY CONTROL?	
9771	25CE	1F	.	RAR ;(RESTORE CHARACTER)	
9772	25CF	FA	1E 26	JM EXP100 ;NO - EXPAND FORMAT CONTROL	
9773	25D2	21	76 FF	LXI H,ENHOUT ;YES - COMPARE TO PREVIOUS	
9774	25D5	AE	.	XRA M ;ANY CHANGES?	
9775	25D6	C8	.	RZ ;NO - RETURN IMMEDIATELY	
9776	25D7	E6	0F .	ANI 170 ;CHANGE IN ENHANCEMENT?	
9777	25D9	CA	F0 25	JZ EXP010 ;NO - CHECK NEW CHARACTER SE	
9778	25DC	06	26 .	MVI B,AMPSND ;YES - OUTPUT ENHANCEMENT	
9779	25DE	CD	FE 14	CALL ECOUTB ;ESCAPE SEQUENCE:	
9780	25E1	3E	64 .	MVI A,SMALLD ;<ESC>--<&>--<LOWER CASE D>	
9781	25E3	CD	04 15	CALL A2OUTB	
9782	25E6	79	.	MOV A,C ;COMPUTE ENHANCEMENT	
9783	25E7	E6	0F .	ANI 170 ;PARAMETER (@-0)	
9784	25E9	F6	40 .	ORI 1000 ;ADJUST TO ASCII LETTER	
9785	25EB	CD	04 15	CALL A2OUTB ;PUT IT INTO OUTPUT BUFFER	
9786	25EE	2E	76 .	MVI L,ENHOUT-BASE ;CHECK CHARACTER SET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 300
9788	25F0	.	.	.	;
9789	25F0	.	.	.	; CHECK FOR CHARACTER SET CHANGE
9790	25F0	.	.	.	;
9791	25F0	.	.	.	EXP010 EQU \$
9792	25F0	79	.	.	MOV A,C ;RECALL CURRENT SETTING
9793	25F1	AE	.	.	XRA M ;COMPARE TO PREVIOUS VALUE
9794	25F2	E6	30	.	ANI 60Q ;ANY CHANGE IN CHAR SET?
9795	25F4	71	.	.	MOV M,C ;(SAVE NEW SETTING)
9796	25F5	C8	.	.	RZ ;NO - RETURN
9797	25F6	79	.	.	MOV A,C ;YES - RECALL NEW SETTING
9798	25F7	E6	30	.	ANI 60Q ;RETURN TO BASE SET?
9799	25F9	CA	19	26	JZ EXP030 ;YES - SEND SHIFT IN (SI)
9800	25FC	2E	75	.	MVI L,CALTST-BASE ;IS IT THE SAME
9801	25FE	BE	.	.	CMP M ;ALTERNATE CHAR SET?
9802	25FF	CA	14	26	JZ EXP020 ;YES - SEND SHIFT OUT ONLY
9803	2602	77	.	.	MOV M,A ;NO - SAVE NEW ALTERNATE
9804	2603	.	.	.	;
9805	2603	.	.	.	; GENERATE ESCAPE SEQUENCE FOR ALTERNATE
9806	2603	.	.	.	; CHARACTER SET SPECIFIER
9807	2603	.	.	.	;
9808	2603	06	29	.	MVI B,ARPARN ;OUTPUT <ESC>
9809	2605	CD	FE	14	CALL ECUOTB ;<RIGHT PARENTHESIS>
9810	2608	79	.	.	MOV A,C
9811	2609	E6	30	.	ANI 60Q ;COMPUTE ALTERNATE CHARACTER
9812	260B	0F	.	.	RRC ;SET PARAMETER
9813	260C	0F	.	.	RRC
9814	260D	0F	.	.	RRC
9815	260E	0F	.	.	RRC
9816	260F	C6	40	.	ADI 100Q
9817	2611	CD	04	15	CALL A2OUTB ;SEND IT
9818	2614	.	.	.	;
9819	2614	.	.	.	EXP020 EQU \$
9820	2614	3E	0E	.	MVI A,SO ;SEND SHIFT OUT (SO)
9821	2616	C3	04	15	JMP A2OUTB ;AND RETURN
9822	2619	.	.	.	;
9823	2619	.	.	.	EXP030 EQU \$
9824	2619	3E	0F	.	MVI A,SI ;SEND SHIFT IN
9825	261B	C3	04	15	JMP A2OUTB ;AND RETURN

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
9827	261E	.	.	;
9828	261E	.	.	; EXPAND ON FORMAT CONTROL
9829	261E	.	.	;
9830	261E	.	.	EXP100 EQU \$
9831	261E	FE	C2	CPI XONLY ;TRANSMIT ONLY CONTROL?
9832	2620	06	7B	MVI B,LFTBRC ;(SET FOR LEFT BRACE)
9833	2622	CA	FE 14	JZ ECOUTB ;YES - OUTPUT AND EXIT
9834	2625	F2	34 26	JP EXP110 ;TYPE DEF - OUTPUT NUMBER
9835	2628	FE	C1	CPI ENDR ;END PROTECT?
9836	262A	06	5B	MVI B,LFTBKT ;(SET FOR LEFT BRACKET - [
9837	262C	CA	FE 14	JZ ECOUTB ;YES - OUTPUT AND EXIT
9838	262F	04	.	INR B ;NO - ALTER CHAR TO RIGHT
9839	2630	04	.	INR B ;BRACKET AND OUTPUT IT
9840	2631	C3	FE 14	JMP ECOUTB
9841	2634	.	.	;
9842	2634	.	.	; TYPE DEFINITION - OUTPUT NUMERIC TERMINATOR
9843	2634	.	.	;
9844	2634	.	.	EXP110 EQU \$
9845	2634	FE	C8	CPI SFKYAT ;IS CODE VALID?
9846	2636	06	7F	MVI B,ADEL ;(SET DEL CHAR FOR INVALID
9847	2638	F2	03 15	JP B2OUTB ;NO - RETURN DEL CHARACTER
9848	263B	FE	C3	CPI FILL ;FILL CODE?
9849	263D	CA	03 15	JZ B2OUTB ;YES - RETURN DEL CHARACTER
9850	2640	D6	8F	SUI ALPHA-6-ZERO ;COMPUTE ASCII DIGIT
9851	2642	47	.	MOV B,A ;PUT CHARACTER INTO B-REG
9852	2643	C3	FE 14	JMP ECOUTB ;OUTPUT THE ESCAPE SEQUENCE

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 302
9854	2646	.	.	.	;
9855	2646	.	.	.	; GET DISPLAY DATA
9856	2646	.	.	.	;
9857	2646	.	.	.	GDS010 EQU \$
9858	2646	CD	E5	1A	CALL CHKSFK ;SOFT KEY MODE?
9859	2649	CA	ED	26	JZ GDS050 ;NO - DO NORMAL PROCEDURE
9860	264C	.	.	.	;!!!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!!!*
9861	264C	CD	23	60	CALL ZMUCHK ;AUTOPLLOT MENU ON?
9862	264F	C2	38	27	JNZ GDS160 ;YES, REPORT END OF DISPLAY
9863	2652	.	.	.	;*****
9864	2652	3A	C0	FF	LDA CURROW ;YES - GET CURSOR ROW
9865	2655	.	.	.	;***** GRAPHICS MODIFICATION *****
9866	2655	FE	12	.	CPI NMFCTK*2 ;BEYOND SOFT KEY DATA?
9867	2657	.	.	.	;*****
9868	2657	F2	38	27	JP GDS160 ;YES - RETURN END OF DISPLAY
9869	265A	0F	.	.	RRC ;IN ATTRIBUTE ROW?
9870	265B	DA	B9	26	JC GDS030 ;NO - OUTPUT DISPLAY DATA
9871	265E	06	26	.	MVI B,AMPSND ;YES - START ESCAPE SEQUENCE
9872	2660	CD	97	7D	CALL INITD1 ;INIT OUTPUT BUFFER POINTERS
9873	2663	CD	FE	14	CALL ECOUTB ;SEND <ESC>-<8>
9874	2666	3E	66	.	MVI A,SMALLF ;<LOWER CASE F>
9875	2668	CD	04	15	CALL A2OUTB
9876	266B	3A	C0	FF	LDA CURROW
9877	266E	0F	.	.	RRC ;<KEY NUMBER>
9878	266F	F6	30	.	ORI ZERO
9879	2671	CD	04	15	CALL A2OUTB
9880	2674	3E	6B	.	MVI A,SMALLK ;<LOWER CASE K>
9881	2676	CD	04	15	CALL A2OUTB
9882	2679	2A	C9	FF	LHLD LSTLIN ;GET ADDRESS OF CURRENT
9883	267C	7D	.	.	MOV A,L ;LINE
9884	267D	D6	08	.	SUI ATBLOC ;COMPUTE LOCATION OF
9885	267F	6F	.	.	MOV L,A ;ATTRIBUTE CODE
9886	2680	7E	.	.	MOV A,M ;GET ATTRIBUTE CODE
9887	2681	06	30	.	MVI B,ZERO ;COMPUTE ATTRIBUTE CODE:
9888	2683	FE	4E	.	CPI N
9889	2685	CA	8F	26	JZ GDS020 ;0 = NORMAL
9890	2688	04	.	.	INR B ;1 = LOCAL ONLY
9891	2689	FE	4C	.	CPI L ;2 = TRANSMIT ONLY
9892	268B	CA	8F	26	JZ GDS020
9893	268E	04	.	.	INR B
9894	268F	.	.	.	GDS020 EQU \$
9895	268F	CD	03	15	CALL B2OUTB ;OUTPUT ATTRIBUTE CODE
9896	2692	3E	61	.	MVI A,SMALLA
9897	2694	CD	04	15	CALL A2OUTB ;OUTPUT <LOWER CASE A>
9898	2697	3E	10	.	MVI A,FRSOUT ;SET FLAG TO INDICATE FIRST
9899	2699	CD	94	18	CALL SETMF2 ;SOFT KEY DATA OUT
9900	269C	CD	3A	1F	CALL FLDSR ;LOCATE THE DATA FIELD
9901	269F	EB	.	.	XCHG
9902	26A0	22	73	FF	SHLD GETADR ;SAVE FIRST CHAR ADDRESS
9903	26A3	.	.	.	; RESTART "GETDSP" TO OUTPUT

13255
2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

=====

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 303		
9904	26A3	.	.	.	;	FIRST SOFT KEY CHAR

=====

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 304
=====
9906      26A3      . . .      ;
9907      26A3      . . .      ; * * * * *
9908      26A3      . . .      ;
9909      26A3      . . .      ; GETDSP - GET A CHARACTER FROM DISPLAY
9910      26A3      . . .      ;
9911      26A3      . . .      ; ENTRY:  CURADR = ADDR OF DISPLAY BYTE
9912      26A3      . . .      ;          CURCOL = COLUMN OF NEXT BYTE
9913      26A3      . . .      ;
9914      26A3      . . .      ; EXIT :  NC - CHARACTER FOUND
9915      26A3      . . .      ;          A = CHARACTER
9916      26A3      . . .      ;          GETADR,CURCOL UPDATED
9917      26A3      . . .      ;          C - NO CHARACTER
9918      26A3      . . .      ;          M - END OF DISPLAY
9919      26A3      . . .      ;          Z - END OF FIELD
9920      26A3      . . .      ;          P,NZ - END OF LINE
9921      26A3      . . .      ;          B-L DESTROYED
9922      26A3      . . .      ;
9923      26A3      . . .      GETDSP EQU $
9924      26A3      . . .      ;*****
9925      26A3      CD 61 60      CALL ZGRTST      ;GETTING GRAPHICS?
9926      26A6      C2 67 60      JNZ ZGRGET      ;YES, GET GRAPHICS DATA
9927      26A9      . . .      ;*****
9928      26A9      21 3C FF      LXI H,B2DPTR    ;GET EXPANSION BUFFER
9929      26AC      7E . .      MOV A,M          ;POINTER
9930      26AD      2B . .      DCX H           ;SET ADDRESS TO END POINTER
9931      26AE      8E . .      CMP M           ;BUFFER EMPTY?
9932      26AF      CA 46 26      JZ GDS010       ;YES - GET BYTE FROM DISPLAY
9933      26B2      2C . .      INR L           ;NO - INCREMENT POINTER AND
9934      26B3      3C . .      INR A           ;STORE IT
9935      26B4      77 . .      MOV M,A
9936      26B5      6F . .      MOV L,A         ;SET BUFFER ADDRESS
9937      26B6      7E . .      MOV A,M         ;GET THE DATA BYTE
9938      26B7      87 . .      ORA A          ;SET C = FALSE
9939      26B8      C9 . .      RET             ;RETURN CHARACTER FOUND
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 305
9941	26B9	.	.	.	;	
9942	26B9	.	.	.	; GET SOFT KEY DATA	
9943	26B9	.	.	.	;	
9944	26B9	.	.	.	GDS030 EQU \$	
9945	26B9	21	6F	FF	LXI H,MFLGS2 ;GET MODE FLAGS	
9946	26BC	7E	.	.	MOV A,M ;MASK OUT FIRST OUTPUT FLAG	
9947	26BD	E6	EF	.	ANI 377Q-FRSOUT	
9948	26BF	BE	.	.	CMP M ;FIRST DATA?	
9949	26C0	CA	ED	26	JZ GDS050 ;NO - GET NEXT DATA	
9950	26C3	77	.	.	MOV M,A ;YES - UPDATE FLAG	
9951	26C4	CD	97	7D	CALL INITD1 ;INITIALIZE CHAR BUFFER PTRS	
9952	26C7	2A	73	FF	LHLD GETADR ;LOCATE END OF LINE	
9953	26CA	EB	.	.	XCHG ;PUT START ADDRESS IN D,E	
9954	26CB	3E	4F	.	MVI A,MAXCOL ;SEARCH TO END OF LINE	
9955	26CD	CD	D6	20	CALL FNDLS0 ;ANY "EOL" IN DATA LINE?	
9956	26D0	3E	50	.	MVI A,MAXCOL+1 ;(SET FOR NO EOL LENGTH=80	
9957	26D2	FA	D6	26	JM GDS040 ;NO - OUTPUT VALUE MAXCOL+1	
9958	26D5	90	.	.	SUB B ;YES - COMPUTE EOL LOCATION	
9959	26D6	.	.	.	GDS040 EQU \$	
9960	26D6	F5	.	.	PUSH PSW ;SAVE DATA LENGTH	
9961	26D7	21	04	15	LXI H,A2OUTB ;SET OUTPUT ROUTINE ADDRESS	
9962	26DA	CD	23	09	CALL BN2DE1 ;CONVERT AND STORE IN BUFFER	
9963	26DD	3E	4C	.	MVI A,L ;OUTPUT UPPER CASE L	
9964	26DF	CD	04	15	CALL A2OUTB	
9965	26E2	F1	.	.	POP PSW ;RECALL DATA LENGTH	
9966	26E3	3D	.	.	DCR A ;DOES DATA EXIST?	
9967	26E4	3E	20	.	MVI A,ABLNK ;(SET TO ADD BLANK)	
9968	26E6	FC	04	15	CM A2OUTB ;NO - ADD A BLANK TO OUTPUT	
9969	26E9	C3	A3	26	JMP GETDSP ;OUTPUT LENGTH PARAMETER	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 306
=====
9971     26EC      . . .      ;
9972     26EC      . . .      ; GET NEXT BYTE FROM DISPLAY
9973     26EC      . . .      ;
9974     26EC      . . .      GDS045 EQU $          ;ENTRY TO SKIP TERMINATOR
9975     26EC      77 . . .      MOV M,A          ;UPDATE "DFLGS" TO CLEAR
9976     26ED      . . .      ;              SKIP TERMINATOR FLAG
9977     26ED      . . .      GDS050 EQU $
9978     26ED      2A 73 FF      LHLD GETADR ;GET CURRENT ADDRESS
9979     26F0      AF . . .      XRA A
9980     26F1      B5 . . .      ORA L          ;END OF DISPLAY?
9981     26F2      CA 35 27      JZ GDS150      ;YES - TERMINATE
9982     26F5      . . .      GDS060 EQU $
9983     26F5      7E . . .      MOV A,M
9984     26F6      2B . . .      DCX H          ;DECREMENT TO NEXT BYTE
9985     26F7      22 73 FF      SHLD GETADR ;UPDATE "GETADR"
9986     26FA      B7 . . .      ORA A          ;IS BYTE ASCII?
9987     26FB      F2 0F 27      JP GDS100      ;YES - RETURN CHARACTER
9988     26FE      FE D0 .      CPI LNKLIM     ;IS IT A LINK?
9989     2700      DA 3C 27      JC GDS200      ;NO - PROCESS DISPLAY CONTRU
9990     2703      6E . . .      MOV L,M        ;YES - SET NEW ADDRESS
9991     2704      67 . . .      MOV H,A
9992     2705      7D . . .      MOV A,L        ;PUT LSB INTO A-REGISTER
9993     2706      2F . . .      CMA
9994     2707      E6 0F .      ANI BLKSM     ;IS IT AN END OF LINE LINK?
9995     2709      CA F5 26      JZ GDS060      ;NO - CONTINUE THRU CHAIN
9996     270C      C3 C1 27      JMP GDS320     ;YES - CHECK TERMINATON
=====

```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 307
=====
 9998     270F      . . .      ;
 9999     270F      . . .      ; ASCII BYTE FOUND - RETURN CHARACTER FOUND
10000     270F      . . .      ;
10001     270F      . . .      GDS100 EQU $
10002     270F      47 . .      MOV B,A          ;SAVE THE CHARACTER
10003     2710      11 C1 FF      LXI D,CURCOL
10004     2713      1A . .      LDAX D          ;INCREMENT CURSOR COLUMN
10005     2714      3C . .      INR A           ;POSITION
10006     2715      12 . .      STAX D
10007     2716      32 00 87      STA IOCRCL      ;UPDATE DISPLAY CURSOR
10008     2719      3A 04 50      LDA BLKTRM     ;GET BLOCK TERMINATOR CHAR
10009     271C      B8 . .      CMP B          ;IS CHAR = BLOCK TERMINATOR?
10010     271D      3E F7 .      MVI A,377Q-SKPTRM ;(SET CLEAR FLAG)
10011     271F      CA 27 27      JZ GDS110      ;YES - RETURN TERMINATION
10012     2722      CD 53 17      CALL CLRDFL    ;NO - CLEAR SKIP FLAG
10013     2725      78 . .      MOV A,B        ;RECALL DISPLAY CHARACTER
10014     2726      C9 . .      RET           ;RETURN (NC FROM "CLRDFL")
10015     2727      . . .      ;*****
10016     2727      . . .      ; BLOCK TERMINATOR - CHECK FOR END OF LINE, *
10017     2727      . . .      ; RETURN END OF DISPLAY *
10018     2727      . . .      ;*****
10019     2727      . . .      GDS110 EQU $
10020     2727      21 6E FF      LXI H,DFLGS    ;CLEAR SKIP TERMINATOR FLAG
10021     272A      A6 . .      ANA M
10022     272B      BE . .      CMP M          ;WAS SKIP FLAG SET?
10023     272C      C2 EC 26      JNZ GDS045     ;YES - IGNORE TERMINATOR
10024     272F      1A . .      LDAX D        ;NO - TERMINATE TRANSMISSION
10025     2730      FE 50 .      CPI MAXCOL+1   ;WAS RS IN LAST COLUMN?
10026     2732      CC 22 22      CZ CRLF       ;YES - DO CR,LF
10027     2735      . . .      ;
10028     2735      . . .      ; RETURN END OF DISPLAY
10029     2735      . . .      ;
10030     2735      . . .      GDS150 EQU $
10031     2735      CD 96 1F      CALL FLDSRX    ;SET "LSTCOL" TO MAXCOL+1 TO
10032     2738      . . .      ; FORCE LINE RE-SCAN
10033     2738      . . .      GDS160 EQU $
10034     2738      AF . .      XRA A         ;SET A TO -1
10035     2739      3D . .      DCR A
10036     273A      37 . .      STC          ;SET C-FLAG TRUE
10037     273B      C9 . .      RET           ;RETURN
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 308
=====
10039    273C      . . .      ;
10040    273C      . . .      ; NON-ASCII BYTE PROCESSING
10041    273C      . . .      ;
10042    273C      . . .      GDS200 EQU $
10043    273C      FE CE      CPI EOP          ;END OF DISPLAY?
10044    273E      CA 35 27   JZ GDS150        ;YES - RETURN END OF DISPLAY
10045    2741      FE C4      CPI STPFLG       ;NON-DISPLAYING TERMINATOR?
10046    2743      CA 89 27   JZ GDS230        ;YES - RETURN END OF DISPLAY
10047    2746      FE CC      CPI EOL          ;END OF LINE?
10048    2748      CA 9F 27   JZ GDS300        ;YES - RETURN END OF LINE
10049    2748      FE C3      CPI FILL         ;FILL BYTE?
10050    274D      CA F5 26   JZ GDS060        ;YES - GET NEXT BYTE
10051    2750      4F .      MOV C,A         ;NO - SAVE THE BYTE
10052    2751      CD CF 1A   CALL CHKFMS     ;FORMAT/SOFT KEY DEFINE MODE
10053    2754      C2 64 27   JNZ GDS210       ;YES - LOOK FOR START PROTEC
10054    2757      3A 64 FF   LDA IOFLG2      ;NO - GET I/O FLAGS 2
10055    275A      E6 20      ANI XDS2BF      ;DISPLAY TO I/O BUFFER?
10056    275C      79 .      MOV A,C         ;(RECALL DATA BYTE)
10057    275D      C0 .      RNZ             ;YES - RETURN UNEXPANDED BYT
10058    275E      CD CA 25   CALL EXPAND     ;NO - EXPAND DISPLAY CONTROL
10059    2761      C3 A3 26   JMP GETDSP      ;RETURN 1ST EXPANDED CHAR
10060    2764      . . .      ;
10061    2764      . . .      ; FORMAT MODE - IGNORE ALL DISPLAY CONTROL EXCEPT
10062    2764      . . .      ; FOR START PROTECT
10063    2764      . . .      ;
10064    2764      . . .      GDS210 EQU $
10065    2764      79 .      MOV A,C         ;RECALL THE DATA BYTE
10066    2765      FE C0      CPI STPR        ;IS IT START PROTECT?
10067    2767      C2 F5 26   JNZ GDS060      ;NO - IGNORE THE BYTE
10068    276A      EB .      XCHG           ;YES - PUT GETADR INTO D,E
10069    276B      2A C0 FF   LHLD CURROW    ;SAVE ENDING ROW AND
10070    276E      3A A3 FF   LDA TLIND      ;COLUMN+1 FOR FIELD
10071    2771      85 .      ADD L
10072    2772      6F .      MOV L,A        ;SAVE ABSOLUTE ROW NUMBER
10073    2773      22 20 FF   SHLD ENDROW
10074    2776      . . .      GDS220 EQU $
10075    2776      CD 7E 11   CALL GTMOD1    ;PAGE MODE/DISPLAY -> BUFFER
10076    2779      CA 86 27   JZ GDS225      ;NO - RETURN END OF FIELD
10077    277C      CD 2F 1F   CALL FLDSR1    ;ANY MORE FIELDS?
10078    277F      CA 35 27   JZ GDS150      ;NO - EXIT END OF DISPLAY
10079    2782      EB .      XCHG           ;YES - STORE NEW GETADR
10080    2783      22 73 FF   SHLD GETADR
10081    2786      . . .      GDS225 EQU $
10082    2786      AF .      XRA A         ;RETURN END OF FIELD
10083    2787      37 .      STC           ;(C = TRUE, A = 0)
10084    2788      C9 .      RET
=====

```

				PAGE 309	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	
10086	2789	.	.	.	;*****
10087	2789	.	.	.	; NON-DISPLAYING TERMINATOR FOUND - CHECK FOR *
10088	2789	.	.	.	; AUTO CLEAR OPTION *
10089	2789	.	.	.	;*****
10090	2789	.	.	.	GDS230 EQU \$
10091	2789	3A	FA	FF	LDA KBJMP2 ;GET JUMPERS SET 2
10092	278C	E6	02	.	ANI CLRTRM ;CLEAR TERMINATOR?
10093	278E	3A	C1	FF	LDA CURCOL ;(SET CURRENT COLUMN)
10094	2791	4F	.	.	MOV C,A
10095	2792	2A	73	FF	LHLD GETADR ;(SET LOCATION OF
10096	2795	23	.	.	INX H ;TERMINATOR)
10097	2796	EB	.	.	XCHG ;(PUT ADDRESS INTO D,E)
10098	2797	C4	11	1C	CNZ CHRDL2 ;YES - CLEAR THE BYTE
10099	279A	EB	.	.	XCHG
10100	279B	2B	.	.	DCX H ;SET LAST CHARACTER ADDRESS
10101	279C	C3	35	27	JMP GDS150 ;RETURN END OF DISPLAY

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 310
10103	279F	.	.	.	;	
10104	279F	.	.	.	; END OF LINE - PAD OUT LINE IF FORMAT MODE	
10105	279F	.	.	.	;	
10106	279F	.	.	.	GDS300 EQU \$	
10107	279F	CD	CF	1A	CALL CHKFMS ;FORMAT/SOFT KEY DEFINE MODE	
10108	27A2	CA	BE	27	JZ GDS310 ;NO - ADVANCE TO NEXT LINE	
10109	27A5	FA	BE	27	JM GDS310 ;SOFT KEY - SKIP TO NEXT LIN	
10110	27A8	11	C1	FF	LXI D,CURCOL ;FORMAT - BLANK FILL	
10111	27AB	1A	.	.	LDAX D ;GET CURRENT CURSOR COLUMN	
10112	27AC	FE	50	.	CPI MAXCOL+1 ;LINE COMPLETED?	
10113	27AE	CA	BE	27	JZ GDS310 ;YES - ADVANCE TO NEXT LINE	
10114	27B1	3C	.	.	INR A ;NO - INCREMENT COLUMN	
10115	27B2	12	.	.	STAX D ;NUMBER	
10116	27B3	32	00	87	STA IUCRCL ;UPDATE DISPLAY CURSOR	
10117	27B6	23	.	.	INX H ;RESTORE "GETADR" TO LOCATIO	
10118	27B7	22	73	FF	SHLD GETADR ;OF "EOL"	
10119	27BA	3E	20	.	MVI A,ABLNK ;RETURN BLANK	
10120	27BC	B7	.	.	ORA A ;CLEAR C-FLAG	
10121	27BD	C9	.	.	RET ;RETURN	
10122	27BE	.	.	.	;	
10123	27BE	.	.	.	GDS310 EQU \$	
10124	27BE	CD	C2	1A	CALL CHAIN1 ;GET ADDR OF NEXT LINE LINK	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 311
=====

```

```

10126  27C1      .      .      .      ;
10127  27C1      .      .      .      ; EOL LINK FOUND - DETERMINE TERMINATION TYPE
10128  27C1      .      .      .      ;
10129  27C1      .      .      .      GDS320 EQU $
10130  27C1      7E      .      .      MOV A,M          ;GET POINTER TO NEXT LINE
10131  27C2      2B      .      .      DCX H
10132  27C3      6E      .      .      MOV L,M
10133  27C4      67      .      .      MOV H,A
10134  27C5      22      73      FF      SHLD GETADR      ;PUT IT INTO "GETADR"
10135  27C8      AF      .      .      XRA A           ;PUT CURSOR IN COLUMN ZERO
10136  27C9      32      76      FF      STA ENHOUT       ;CLEAR LAST ENHANCE OUT FLAG
10137  27CC      CD      7F      23      CALL CRRET1
10138  27CF      CD      CF      1A      CALL CHKFMS      ;FORMAT/SOFT KEY DEFINE MODE
10139  27D2      CA      1B      7D      JZ GDS360        ;NEITHER - SEND END OF LINE
10140  27D5      FA      15      7D      JM GDS350        ;SOFT KEY - FIND NEXT FIELD
10141  27D8      2A      C0      FF      LHLD CURROW     ;FORMAT - SAVE ENDING ROW AN
10142  27DB      3A      A3      FF      LDA TLINO       ;COLUMN+1 FOR FIELD
10143  27DE      85      .      .      ADD L
10144  27DF      6F      .      .      MOV L,A         ;SAVE ABSOLUTE ROW NUMBER
10145  27E0      22      20      FF      SHLD ENDROW
10146  27E3      CD      6F      0B      CALL LNFEED     ;YES - DO LINE FEED
10147  27E6      CD      03      10      CALL DISLN1     ;SET DISPLAY CURSOR ROW
10148  27E9      2A      73      FF      LHLD GETADR     ;RECALL POINTER TO NEXT LINE
10149  27EC      7D      .      .      MOV A,L         ;GET LSB VALUE
10150  27ED      B7      .      .      ORA A           ;END OF DISPLAY (LSB = 0)?
10151  27EE      CA      38      27      JZ GDS160       ;YES - RETURN END OF DISPLAY
10152  27F1      .      .      .      ;*****
10153  27F1      .      .      .      ; MOVE MAIN CODE OVERFLOW TO SECOND ROM BOARD
10154  27F1      C3      00      7D      JMP OVRFLO
10155  27F4      .      .      .      ORG 76400Q     ;BASE OF OVERFLOW CODE
10156  7D00      .      .      .      OVRFLO EQU $
10157  7D00      .      .      .      ;*****
10158  7D00      .      .      .      ;
10159  7D00      .      .      .      ; FORMAT EOL - CHECK FOR CONTINUATION FIELD
10160  7D00      .      .      .      ;
10161  7D00      EB      .      .      XCHG           ;PUT CURRENT ADDR IN D,E
10162  7D01      CD      FA      1F      CALL FLDSR2     ;NEXT LINE CONTINUES FIELD?
10163  7D04      C2      76      27      JNZ GDS220     ;NO - RETURN END OF FIELD
10164  7D07      EB      .      .      XCHG           ;YES - CONTINUE PROCESSING
10165  7D08      3A      64      FF      LDA IOFLG2     ;GET I/O FLAGS 2
10166  7D0B      E6      20      .      ANI XDS2BF     ;DISPLAY TO I/O BUFFER?
10167  7D0D      CA      F5      26      JZ GDS060     ;YES - CONTINUE FIELD
10168  7D10      22      73      FF      SHLD GETADR     ;NO - STORE NEW "GETADR"
10169  7D13      37      .      .      STC           ;RETURN END OF LINE
10170  7D14      C9      .      .      RET           ;RETURN NZ, C

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 312
=====
10172    7D15    . . .      ;*****
10173    7D15    . . .      ; END OF LINE FOR NON-FORMAT MODE - RETURN END *
10174    7D15    . . .      ;   OF LINE CODE (C, P, NZ)                          *
10175    7D15    . . .      ;*****
10176    7D15    . . .      GDS350 EQU $           ;SOFT KEY END OF LINE
10177    7D15    CD 6F 0B    CALL LNFEED           ;LOCATE THE ATTRIBUTE OF THE
10178    7D18    CD 3A 1F    CALL FLDSR            ;NEXT DEFINITION
10179    7D1B    . . .      GDS360 EQU $           ;NON-FORMAT/SOFT KEY EOL
10180    7D1B    AF . .     XRA A                 ;SET NZ,P
10181    7D1C    3C . .     INR A
10182    7D1D    37 . .     STC                    ;SET C-TRUE
10183    7D1E    C9 . .     RET                    ;RETURN END OF LINE
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 313
10185	7D1F	.	.	;*****	
10186	7D1F	.	.	; INITDG - INITIALIZE FOR DISPLAY GET *	
10187	7D1F	.	.	;*****	
10188	7D1F	.	.	;	
10189	7D1F	.	.	; EXIT : Z - CHARACTER FOUND	
10190	7D1F	.	.	; GETADR = ADDRESS OF FIRST CHARACTER	
10191	7D1F	.	.	; NZ - NO CHARACTER FOUND	
10192	7D1F	.	.	; GETADR UNCHANGED	
10193	7D1F	.	.	; ALL REGISTERS DESTROYED	
10194	7D1F	.	.	;	
10195	7D1F	.	.	; DISPLAY GET ROUTINE IS SET TO START	
10196	7D1F	.	.	; AT CURRENT CURSOR LOCATION	
10197	7D1F	.	.	;	
10198	7D1F	.	.	INITD0 EQU \$;ENTRY FOR DISPLAY TO I/O	
10199	7D1F	.	.	;*****	
10200	7D1F	CD	61 60	CALL ZGRST ;INITIALIZE FOR GRAPHICS GET	
10201	7D22	C2	2E 7D	JNZ INITDG ;YES, DONT MOVE CURSOR	
10202	7D25	.	.	;*****	
10203	7D25	CD	D4 1A	CALL CHKfmt ;FORMAT MODE ENABLED?	
10204	7D28	C2	2E 7D	JNZ INITDG ;YES - DON'T MOVE CURSOR	
10205	7D2B	32	C1 FF	STA CURCOL ;NO - BEGIN AT LINE START	
10206	7D2E	.	.	INITDG EQU \$	
10207	7D2E	CD	6A 18	CALL SETDF0 ;SET DATA COMM INPUT FLAG TO	
10208	7D31	.	.	ENABLE TRANSMIT ONLY DATA	
10209	7D31	E6	FB .	ANI 377Q-NOSEND ;CLEAR NO DATA FLAG	
10210	7D33	BE	. .	CMP M ;WAS IT SET BEFORE?	
10211	7D34	77	. .	MOV M,A ;(SET NEW VALUE)	
10212	7D35	C0	. .	RNZ ;YES - RETURN NO DATA	
10213	7D36	2E	64 .	MVI L,IOFLG2-BASE ;CLEAR DISPLAY BOUNDARY	
10214	7D38	7E	. .	MOV A,M ;FLAGS	
10215	7D39	E6	3F .	ANI 377Q-ENDDSP-DSPBTM	
10216	7D38	77	. .	MOV M,A	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 314
=====
10218    7D3C      . . .      ;
10219    7D3C      . . .      ; LOCATE FIRST CHARACTER
10220    7D3C      . . .      ;
10221    7D3C      CD E5 1A    CALL CHKSFK      ;SOFT KEY MODE?
10222    7D3F      . . .      ;*****
10223    7D3F      CA 9E 7D    JZ  IDG110      ;NO, TEST FOR GRAPHICS INIT
10224    7D42      . . .      ;*****
10225    7D42      21 C0 FF    LXI  H,CURROW   ;YES - CHECK CURSOR POSITION
10226    7D45      3E 10 .     MVI  A,SFTEND
10227    7D47      BE . . .    CMP  M          ;CURSOR BELOW DATA AREA?
10228    7D48      F8 . . .    RM             ;YES - RETURN NO CHARACTER
10229    7D49      3E FE .     MVI  A,376Q    ;NO - SET CURSOR ROW TO
10230    7D4B      A6 . . .    ANA  M          ;ATTRIBUTE ROW
10231    7D4C      77 . . .    MOV  M,A
10232    7D4D      AF . . .    XRA  A          ;SET CURSOR COLUMN TO
10233    7D4E      23 . . .    INX  H          ;BEGINNING OF ROW
10234    7D4F      77 . . .    MOV  M,A
10235    7D50      . . .      ; LOCATE ATTRIBUTE
10236    7D50      . . .      ;
10237    7D50      . . .      IDG055 EQU $
10238    7D50      3E 01 .     MVI  A,IGNTRM  ;SET TO IGNORE NON-DISPLAYIN
10239    7D52      32 6D FF    STA  TRMFACT   ;TERMINATORS
10240    7D55      CD CA 07    CALL RCADR4    ;DISPLAY PRESENT?
10241    7D58      F8 . . .    RM             ;NO - RETURN NO CHARACTER
10242    7D59      CA 65 7D    JZ  IDG060     ;CHARACTER - CHECK PROTECTED
10243    7D5C      CD D4 1A    CALL CHKFMT    ;EOL - FORMAT MODE?
10244    7D5F      CA 80 7D    JZ  IDG100     ;NO - EXIT WITH EOL
10245    7D62      C3 69 7D    JMP  IDG070    ;YES - CHECK PROTECTED
10246    7D65      . . .      ;
10247    7D65      . . .      IDG060 EQU $
10248    7D65      21 C2 FF    LXI  H,PROFLD  ;SET PROTECT STATUS
10249    7D68      70 . . .    MOV  M,B
10250    7D69      . . .      IDG070 EQU $
10251    7D69      CD 90 11    CALL CKPROT    ;CURSOR IN PROTECTED FIELD?
10252    7D6C      C2 7A 7D    JNZ  IDG090    ;NO - RETURN CHARACTER FOUND
10253    7D6F      3E FF .     MVI  A,STPXHR  ;SET TERMINATOR FUNCTION TO
10254    7D71      32 6D FF    STA  TRMFACT   ;TERMINATE TRANSFER
10255    7D74      CD 2F 1F    CALL FLDSR1    ;ANY MORE FIELDS?
10256    7D77      CA 0A 0C    JZ  NZEXIT     ;NO - RETURN NO CHARACTER
10257    7D7A      . . .      IDG090 EQU $
10258    7D7A      21 FF 00    LXI  H,377Q    ;INITIALIZE PREVIOUS FIELD'S
10259    7D7D      22 20 FF    SHLD ENDROW    ;ROW AND COLUMN TO ZERO
=====

```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 315
10261	7D80	. . .	;	
10262	7D80	. . .	; CHARACTER FOUND - RETURN CHARACTER FOUND	
10263	7D80	. . .	;	
10264	7D80	. . .	IDG100 EQU \$	
10265	7D80	1A . .	LDAX D ;GET FIRST CHARACTER	
10266	7D81	FE C4 .	CPI STPFLG ;NON-DISPLAYING TERMINATOR?	
10267	7D83	CC 90 0C	CZ NXTCHR ;YES - GET THE NEXT CHARACTE	
10268	7D86	EB . .	XCHG ;SAVE ADDRESS OF BYTE	
10269	7D87	22 73 FF	SHLD GETADR	
10270	7D8A	3A 2A 48	LDA ALTOUT ;SET CURRENT ALTERNATE CHAR	
10271	7D8D	32 75 FF	STA CALTST ;SET TO DEFAULT VALUE	
10272	7D90	3A C6 FF	LDA LSTDCD ;SET LAST ENHANCEMENT OUT	
10273	7D93	32 76 FF	STA ENHOUT ;WORD	
10274	7D96	BF . .	CMP A ;SET Z-FLAG TRUE	
10275	7D97	. . .	INITD1 EQU \$;INITIALIZE CHARACTER BUFFER	
10276	7D97	21 3C 3C	LXI H,B2DBFL-1*256+B2DBFL-1 ;POINTERS	
10277	7D9A	22 3B FF	SHLD B2DEND	
10278	7D9D	C9 . .	RET ;RETURN	
10279	7D9E	. . .	;*****	
10280	7D9E	. . .	IDG110 EQU \$	
10281	7D9E	CD 61 60	CALL ZGRTST ;GOING TO DO GRAHICS GET?	
10282	7DA1	C2 64 60	JNZ ZGGINT ;YES, DO GRAPHICS GET INIT	
10283	7DA4	C3 50 7D	JMP IDG055 ;NO, INIT FUR NORMAL DSPLY	
10284	7DA7	. . .	;*****	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 316
=====
10286    7DA7      . . .      ;*****
10287    7DA7      . . .      ; STAT2 - SEND SECONDARY TERMINAL STATUS REQUEST *
10288    7DA7      . . .      ;*****
10289    7DA7      . . .      STAT2 EQU $
10290    7DA7      01 00 04    LXI B,SSTAT2 ;SET SECONDARY STATUS PENDIN
10291    7DAA      C3 25 18    JMP SBLXF0   ;FLAG
10292    7DAD      . . .      ;*****
10293    7DAD      . . .      ; STA2GO - TRANSMIT SECONDARY TERMINAL STATUS *
10294    7DAD      . . .      ;*****
10295    7DAD      . . .      STA2GO EQU $
10296    7DAD      01 FF FB    LXI B,-1-SSTAT2
10297    7DB0      CD 9B 11    CALL CLBLXF ;CLEAR STATUS 2 PENDING FLAG
10298    7DB3      06 7C .     MVI B,VRTBAR ;SEND <ESC>-<VERTICAL BAR>
10299    7DB5      CD 1C 19    CALL ESCOUT
10300    7DB8      21 22 19    LXI H,XPUTDC ;SET OUTPUT ROUTINE ADDRESS
10301    7DBB      CD C1 7D    CALL STA2G1 ;OUTPUT SECONDARY STATUS BIT
10302    7DBE      C3 4E 13    JMP SDTERM  ;SEND TERMINATOR AND RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 317
10304	7DC1	.	.	*****	
10305	7DC1	.	.	; STA2G1 - OUTPUT SECONDARY STATUS BITS *	
10306	7DC1	.	.	*****	
10307	7DC1	.	.	;	
10308	7DC1	.	.	; ENTRY: H,L = ADDRESS OF OUTPUT ROUTINE	
10309	7DC1	.	.	;	
10310	7DC1	.	.	; EXIT : ALL REGISTER DESTROYED	
10311	7DC1	.	.	; CNTFAD DESTROYED	
10312	7DC1	.	.	;	
10313	7DC1	.	.	STA2G1 EQU \$	
10314	7DC1	22	CE FF	SHLD CNTFAD ;SET OUTPUT ROUTINE ADDRESS	
10315	7DC4	.	.	STA2G2 EQU \$	
10316	7DC4	.	.	;	
10317	7DC4	.	.	; SEND NON-DISPLAY RAM SIZE (K)	
10318	7DC4	.	.	;	
10319	7DC4	3E	D0 .	MVI A,BFSPCE+1/256	
10320	7DC6	21	8E FF	LXI H,BUFBN+1 ;COMPUTE NON-DISPLAY RAM	
10321	7DC9	96	.	SUB M ;SIZE	
10322	7DCA	CD	A6 0C	CALL PAROT2 ;SEND NON-DISPLAY RAM SIZE	
10323	7DCD	.	.	;	
10324	7DCD	.	.	; OUTPUT TERMINAL TYPE	
10325	7DCD	.	.	;	
10326	7DCD	3A	FD FF	LDA TRMTYP ;GET THE TERMINAL TYPE NUMBE	
10327	7DD0	CD	A8 0C	CALL PAROUT ;SEND ONLY LOWER FOUR BITS	
10328	7DD3	.	.	;	
10329	7DD3	.	.	; OUTPUT REMAINING KYBD INTFACE STRAPS	
10330	7DD3	.	.	;	
10331	7DD3	2A	F9 FF	LHLD KBJMP3 ;GET JUMPERS J-Z	
10332	7DD6	7C	.	MOV A,H ;SEND STRAPS J-K-L-M	
10333	7DD7	CD	A8 0C	CALL PAROUT	
10334	7DDA	7C	.	MOV A,H ;SEND STRAPS N-P-Q-R	
10335	7DD8	CD	A4 0C	CALL PAROT4	
10336	7DDE	7D	.	MOV A,L ;SEND STRAPS S-T-U-V	
10337	7DDF	CD	A8 0C	CALL PAROUT	
10338	7DE2	7D	.	MOV A,L ;SEND STRAPS W-X-Y-Z	
10339	7DE3	CD	A4 0C	CALL PAROT4	
10340	7DE6	.	.	;	
10341	7DE6	.	.	; OUTPUT MEMORY LOCK STATUS	
10342	7DE6	.	.	;	
10343	7DE6	3A	6A FF	LDA MLKFLG ;GET MEMORY LOCK FLAG	
10344	7DE9	21	F4 FF	LXI H,MDFLG1 ;COMBINE WITH MODE FLAG	
10345	7DEC	A6	.	ANA M ;EXTRACT MEMORY LOCK STATE	
10346	7DED	E6	04 .	ANI MEMLOK	
10347	7DEF	C3	A8 0C	JMP PAROUT ;OUTPUT MEMORY LOCK STATE	
10348	7DF2	.	.	AND RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 318
=====
10350    7DF2      . . .      ;*****
10351    7DF2      . . .      ; SOFT KEY DATA DONE TABLE - IGNORE DC3,CR,& LF *
10352    7DF2      . . .      ;*****
10353    7DEF      . . .      DFSTB3 EQU $-3
10354    7DF2      0A 0A .      DB 12Q,12Q ;LINE FEED
10355    7DF4      59 A3 .      DW DFS350+B15 ;CHECK FOR IGNORE
10356    7DF6      0D 0D .      DB 15Q,15Q ;RETURN
10357    7DF8      60 A3 .      DW DFS360+B15 ;CHECK FOR IGNORE
10358    7DFA      13 13 .      DB 23Q,23Q ;DC3
10359    7DFC      42 85 .      DW ESCAP1+B15 ;IGNORE IT
10360    7DFE      . . .      ;*****
10361    7DFE      . . .      ; SOFT KEY MODE ENABLED RANGE TABLE *
10362    7DFE      . . .      ;*****
10363    7DFB      . . .      DFSTB0 EQU $-3
10364    7DFE      20 7F .      DB 40Q,177Q ;DISPLAYABLE CHARACTER
10365    7E00      C6 8D .      DW SFKYDS+B15 ;DISPLAY IN PROPER DISPLAY
10366    7E02      . . .      ;*****
10367    7E02      . . .      ; NORMAL CHARACTER SET ATTRIBUTES *
10368    7E02      . . .      ;*****
10369    7DFF      . . .      RTABLE EQU $-3
10370    7E02      20 7F .      DB 40Q,177Q ;ALPHANUMERICS
10371    7E04      83 A5 .      DW DSPCHR+B15 ;DISPLAYABLE CHARACTERS
10372    7E06      07 0F .      DB 7Q,17Q ;BELL,BS,HT,LF,VT,FF,CR,SO,S
10373    7E08      1A 7E .      DW RTB010 ;USE FUNCTION TABLE
10374    7E0A      1B 1B .      DB 33Q,33Q ;ESCAPE
10375    7E0C      11 85 .      DW ESCAPE+B15 ;USE <ESC> RANGE TABLE
10376    7E0E      . . .      ;
10377    7E0E      . . .      ;
10378    7E0E      . . .      ;!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!*
10379    7E0E      1D 1D .      DB 35Q,35Q ;GROUP SEPARATOR
10380    7E10      2F E0 .      DW ZTKSUP+B15 ;SET UP FOR TEK
10381    7E12      1F 1F .      DB 37Q,37Q ;US
10382    7E14      3E E0 .      DW ZTKCLR+B15 ;CLEAR ECHOPLEX SUPRESS
10383    7E16      . . .      ;*****
10384    7E16      . . .      ;
10385    7E16      00 7F .      DB 0Q,177Q ;ALL OTHER CODES
10386    7E18      BB 84 .      DW CHKCTL+B15 ;CHECK FOR BLOCK XFR CHARS
10387    7E1A      . . .      ;
10388    7E1A      . . .      ; <BELL> THROUGH <SHIFT IN>
10389    7E1A      . . .      ;
10390    7E1A      . . .      RTB010 EQU $
10391    7E1A      14 48 .      DW ZBELL ;BELL - SOUND KEYBOARD BELL
10392    7E1C      . . .      RTB020 EQU $ ;<BS> THROUGH <SHIFT IN>
10393    7E1C      77 21 .      DW BCKSPC ;BS - BACKSPACE CURSOR
10394    7E1E      00 21 .      DW HTAB ;HORIZONTAL TAB
10395    7E20      . . .      ;
10396    7E20      . . .      ;*****
10397    7E20      76 60 .      DW ZAPLF ;LINE FEED
10398    7E22      44 60 .      DW ZVT ;VERTICAL TAB
10399    7E24      . . .      ;*****
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 319
=====
10400    7E24      . . .      ;
10401    7E24      CA 1A .      DW  NOFNCT      ;FF - NO FUNCTION
10402    7E26      . . .      ;
10403    7E26      . . .      ;*****
10404    7E26      56 60 .      DW  ZAPCR      ;CARRIAGE RETURN
10405    7E28      . . .      ;*****
10406    7E28      EA 0D .      DW  SHFTOT    ;SHIFT OUT
10407    7E2A      FD 0D .      DW  SHFTIN    ;SHIFT IN
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 320
10409	7E2C	.	.	;*****	
10410	7E2C	.	.	; ESCAPE CHARACTER ATTRIBUTES FOR SOFT KEYS *	
10411	7E2C	.	.	;*****	
10412	7E29	.	.	SESECTB EQU \$-3	
10413	7E2C	29	3E	DB 51Q,76Q ;(<)> TO (>)	
10414	7E2E	48	85	DW ESCEND+B15 ;ABORT ESCAPE SEQUENCE	
10415	7E30	4C	4F	DB 114Q,117Q ;<L> TO <O>	
10416	7E32	48	85	DW ESCEND+B15 ;ABORT ESCAPE SEQUENCE	
10417	7E34	53	58	DB 123Q,130Q ;<S> TO <W>	
10418	7E36	48	85	DW ESCEND+B15 ;ABORT ESCAPE SEQUENCE	
10419	7E38	.	.	; *** LOWER CASE CHARACTERS ***	
10420	7E38	6C	6D	DB 154Q,155Q ;<L> TO <M>	
10421	7E3A	48	85	DW ESCEND+B15 ;ABORT ESCAPE SEQUENCE	
10422	7E3C	79	7B	DB 171Q,173Q ;<Y> TO <I>	
10423	7E3E	48	85	DW ESCEND+B15 ;ABORT ESCAPE SEQUENCE	
10424	7E40	.	.	;*****	
10425	7E40	.	.	; NORMAL ESCAPE CHARACTER ATTRIBUTES *	
10426	7E40	.	.	;*****	
10427	7E3D	.	.	ESCTAB EQU \$-3	
10428	7E40	.	.	;*****	
10429	7E40	2A	2A	DB 52Q,52Q ;<*>	
10430	7E42	08	E0	DW ZGSTUP+B15 ;GRAPHICS ESCAPE SEQUENCE	
10431	7E44	.	.	;*****	
10432	7E44	.	.	;	
10433	7E44	26	26	DB 46Q,46Q ;<8> - AMPERSAND	
10434	7E46	E0	97	DW PRMSEQ+B15 ;PARAMETERIZED SEQUENCE	
10435	7E48	.	.	;	
10436	7E48	.	.	;	
10437	7E48	29	29	DB 51Q,51Q ;) - SPECIFY ALT CHAR SET	
10438	7E4A	85	8D	DW SCHRST+B15	
10439	7E4C	31	35	DB 61Q,65Q ;<1> TO <5>	
10440	7E4E	78	7E	DW EI1	
10441	7E50	36	38	DB 66Q,70Q ;<6> TO <8>	
10442	7E52	E0	90	DW TYPSET+B15 ;DEFINE FIELD TYPE	
10443	7E54	.	.	;	
10444	7E54	3C	3E	DB 74Q,76Q ;(<) TO (>)	
10445	7E56	82	7E	DW EI1A ;USE INDEX TABLE	
10446	7E58	.	.	;	
10447	7E58	40	6D	DB 100Q,155Q ;<@> TO <LOWER CASE M>	
10448	7E5A	88	7E	DW EI2 ;USE INDEX TABLE	
10449	7E5C	.	.	; *** LOWER CASE RANGE ***	
10450	7E5C	78	7B	DB 170Q,173Q ;<X> TO <LEFT BRACE>	
10451	7E5E	E4	7E	DW EI3 ;USE INDEX TABLE	
10452	7E60	.	.	;	
10453	7E60	7E	7E	DB 176Q,176Q ;<^> (TILDE)	
10454	7E62	A7	FD	DW STAT2+B15 ;TERMINAL STATUS 2	
10455	7E64	.	.	;	
10456	7E64	.	.	;	
10457	7E64	.	.	;!!!!!!!!!!!! GRAPHICS MODIFICATION !!!!!!!!!!!!!*	
10458	7E64	0C	0C	DB 14Q,14Q ;FF	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 321
=====
10459     7E66     32  E0  .      DW      ZPAGE+B15 ;TEK 'PAGE'
10460     7E68     1A  1A  .      DB      32Q,32Q  ;SUB
10461     7E6A     35  E0  .      DW      ZSTGIN+B15 ;TEK GIN MODE
10462     7E6C     05  05  .      DB      5Q,5Q    ;ENQ
10463     7E6E     3B  E0  .      DW      ZTKCUR+B15 ;READ TEK CURSOR POSITION
10464     7E70     17  17  .      DB      27Q,27Q  ;ETB
10465     7E72     38  E0  .      DW      ZTKHC+B15 ;MAKE TEK HARD COPY
10466     7E74     .   .   .      ;*****
10467     7E74     .   .   .      ;
10468     7E74     00  7F  .      DB      0Q,177Q  ;ALL OTHER CODES
10469     7E76     48  85  .      DW      ESCEND+B15 ;ABORT ESCAPE SEQUENCE
=====
```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 323
10521	7EC8	. . .	;	
10522	7EC8	. . .	;	LOWER CASE RANGE FOR 2 CHARACTER ESC SEQUENCES
10523	7EC8	. . .	;	
10524	7EC8	02 13 .	DW RLCRSN	;@ - SCREEN RELATIVE SENSE
10525	7ECA	0A 13 .	DW CURSEN	;A - ABSOLUTE CURSOR SENSE
10526	7ECC	4C 17 .	DW KBEN1	;B - ENABLE KEYBOARD
10527	7ECE	59 17 .	DW KBLOK0	;C - DISABLE (LOCK) KEYBOARD
10528	7ED0	22 18 .	DW ENTREN	;D - SEND DISPLAY TO CPU
10529	7ED2	BA 16 .	DW IOBNGO	;E - FAST BINARY READ
10530	7ED4	71 13 .	DW DISMDM	;F - DISCONNECT MODEM
10531	7ED6	DE 0D .	DW SFTRST	;G - SOFT RESET
10532	7ED8	9D 1E .	DW CURPH	;H - HOME TO UNPROTECTED
10533	7EDA	58 1A .	DW BKTAB	;I - BACK TAB
10534	7EDC	AE 0D .	DW SFKYON	;J - TURN ON SOFT KEY MENU
10535	7EDE	96 0D .	DW SFKYOF	;K - RESTORE NORMAL DISPLAY
10536	7EE0	D4 0B .	DW MLKON	;L - MEMORY LOCK ON
10537	7EE2	C9 0B .	DW MLKOFF	;M - MEMORY LOCK OFF
10538	7EE4	. . .	;	
10539	7EE4	. . .	EI3 EQU \$;LOWER CASE <X> TO <[>
10540	7EE4	CA 13 .	DW DCTEST	;X - DATA COMM SELF-TEST
10541	7EE6	B4 15 .	DW MNMDON	;Y - MONITOR MODE ON
10542	7EE8	8E 0E .	DW TEST	;Z - SELF-TEST
10543	7EEA	EB 17 .	DW STRXMO	;[- START TRANSMIT-ONLY

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 324
=====
10545    7EEC      . . .      ;*****
10546    7EEC      . . .      ; PRMTAB - TABLE FOR SEQUENCES WITH PARAMETERS *
10547    7EEC      . . .      ;*****
10548    7EE9      . . .      PRMTAB EQU $-3
10549    7EEC      61 67 .      DB 141Q,147Q ;LOWER CASE <A> TO <G>
10550    7EEE      00 7F .      DW PRM010 ;USE INDEX TABLE
10551    7EF0      . . .      ;
10552    7EF0      68 68 .      DB 153Q,153Q ;LOWER CASE <K>
10553    7EF2      20 C8 .      DW ZSTLKY+B15 ;GO TO SET KEYS ROUTINE
10554    7EF4      . . .      ;
10555    7EF4      70 70 .      DB 160Q,160Q ;LOWER CASE <P>
10556    7EF6      D2 96 .      DW IOCTGO+B15 ;GO TO I/O CONTROL ROUTINE
10557    7EF8      . . .      ;
10558    7EF8      73 73 .      DB 163Q,163Q ;LOWER CASE <S>
10559    7EFA      1D C8 .      DW ZSTJPR+B15 ;GO TO SET JUMPERS ROUTINE
10560    7EFC      00 7F .      DB 0Q,177Q ;ALL OTHER CODES
10561    7EFE      48 85 .      DW ESCEND+B15 ;ABORT ESCAPE SEQUENCE
10562    7F00      . . .      ;
10563    7F00      . . .      PRM010 EQU $ ;LOWER CASE <A> TO <F>
10564    7F00      6B 12 .      DW CURPOS ;A - CURSOR POSITIONING
10565    7F02      63 17 .      DW LOADR ;B - BINARY LOADER
10566    7F04      6E 17 .      DW LOADR1 ;C - LOADER SANS MESSAGE
10567    7F06      32 24 .      DW DISPEN ;D - DISPLAY ENHANCEMENT
10568    7F08      48 05 .      DW ESCEND ;E - INVALID, ABORT SEQUENCE
10569    7F0A      64 22 .      DW DFSFKY ;F - DEFINE FUNCTION KEYS
10570    7F0C      A7 18 .      DW SNDCDE ;G - SEND ATTENTION/FUNCTION
10571    7F0E      . . .      ; CODE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 325
10573	7F0E	.	.	*****	
10574	7F0E	.	.	; DENTAB - DISPLAY ENHANCEMENT ESCAPE TABLE *	
10575	7F0E	.	.	*****	
10576	7F0B	.	.	DENTAB EQU \$-3	
10577	7F0E	40	4F	DB 100Q,117Q ;<@>-<O>	
10578	7F10	3C	A4	DW DISPLC+B15 ;TURN ON ENHANCEMENT	
10579	7F12	.	.	*****	
10580	7F12	.	.	; CHRSTB - ALTERNATE CHARACTER SET TABLE *	
10581	7F12	.	.	*****	
10582	7F0F	.	.	CHRSTB EQU \$-3	
10583	7F12	40	43	DB 100Q,103Q ;<@> - <C>	
10584	7F14	8B	8D	DW SCHST1+B15 ;SET ALTERNATE CHAR SET	
10585	7F16	.	.	;	
10586	7F16	00	7F	DB 0Q,177Q ;ALL OTHER CODES	
10587	7F18	48	85	DW ESCEND+B15 ;ABORT ESCAPE SEQUENCE	
10588	7F1A	.	.	*****	
10589	7F1A	.	.	; CRPTAB - CURSOR POSITIONING ESCAPE TABLE *	
10590	7F1A	.	.	*****	
10591	7F17	.	.	CRPTAB EQU \$-3	
10592	7F1A	2B	2B	DB 53Q,53Q ;<+> - PLUS SIGN	
10593	7F1C	87	93	DW DCPLUS+B15 ;SET SIGN FLAG TO +1	
10594	7F1E	2D	2D	DB 55Q,55Q ;NEGATIVE REL. POSITIONING	
10595	7F20	BC	93	DW DCMNUS+B15 ;SET SIGN FLAG TO -1	
10596	7F22	30	39	DB 60Q,71Q ;VALID PARAMETER DIGITS	
10597	7F24	93	93	DW DCNUM+B15 ;ACCUMULATE NUMERICAL VALUE	
10598	7F26	.	.	;	
10599	7F26	43	43	DB 103Q,103Q ;<C>	
10600	7F28	80	92	DW CURP01+B15 ;SET COLUMN PARAMETER	
10601	7F2A	.	.	;	
10602	7F2A	52	52	DB 122Q,122Q ;<R>	
10603	7F2C	96	92	DW CURP03+B15 ;SET ROW PARAMETER	
10604	7F2E	.	.	;	
10605	7F2E	59	59	DB 131Q,131Q ;<Y>	
10606	7F30	8B	92	DW CURP02+B15 ;SET SCREEN ROW PARAMETER	
10607	7F32	.	.	;	
10608	7F32	63	63	DB 143Q,143Q ;<LOWER CASE C>	
10609	7F34	80	92	DW CURP01+B15 ;SET COLUMN PARAMETER	
10610	7F36	.	.	;	
10611	7F36	72	72	DB 162Q,162Q ;<LOWER CASE R>	
10612	7F38	96	92	DW CURP03+B15 ;SET ROW PARAMTER	
10613	7F3A	.	.	;	
10614	7F3A	79	79	DB 171Q,171Q ;<LOWER CASE Y>	
10615	7F3C	8B	92	DW CURP02+B15 ;SET SCREEN ROW PARAMETER	
10616	7F3E	.	.	;	
10617	7F3E	20	20	DB 40Q,40Q ;SPACE - IGNORE	
10618	7F40	42	85	DW ESCAP1+B15	
10619	7F42	00	7F	DB 0,177Q ;INVALID	
10620	7F44	48	85	DW ESCEND+B15	

13255

2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 326
=====
10622     7F46     . . .      ;*****
10623     7F46     . . .      ; FUNCTION DISABLE ATTRIBUTES *
10624     7F46     . . .      ;*****
10625     7F43     . . .      FDISTB EQU $-3
10626     7F46     0D 0D     .      DB 15Q,15Q ;RETURN CODE
10627     7F48     27 91     .      DW CARRET+B15
10628     7F4A     1B 1B     .      DB 33Q,33Q ;ESCAPE
10629     7F4C     EE 95     .      DW FDESC+B15
10630     7F4E     5A 5A     .      DB 132Q,132Q
10631     7F50     CC 95     .      DW FDISOF+B15
10632     7F52     00 7F     .      DB 0,177Q ;ALL OTHER CODES
10633     7F54     C6 8D     .      DW SFKYDS+B15 ;ADD CHARACTER TO DISPLAY
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 327
10635	7F56	.	.	*****	
10636	7F56	.	.	; BINARY LOADER CHARACTER ATTRIBUTES *	
10637	7F56	.	.	*****	
10638	7F53	.	.	LDRTAB EQU \$-3	
10639	7F56	41	46	DB 101Q,106Q ;<A> - <F>	
10640	7F58	7E	7F	DW LI1 ;USE INDEX TABLE	
10641	7F5A	.	.	;	
10642	7F5A	61	64	DB 141Q,144Q ;LOADER COMMAND	
10643	7F5C	7E	7F	DW LI1 ;USE INDEX TABLE	
10644	7F5E	.	.	;	
10645	7F5E	0A	0A	DB 12Q,12Q ;LINE FEED	
10646	7F60	42	85	DW ESCAP1+B15	
10647	7F62	0D	0D	DB 15Q,15Q ;CR	
10648	7F64	42	85	DW ESCAP1+B15	
10649	7F66	13	13	DB 23Q,23Q ;DC3	
10650	7F68	42	85	DW ESCAP1+B15 ;IGNORE	
10651	7F6A	.	.	*****	
10652	7F6A	.	.	; SNDCTB - ACCUMULATE ATTENTION/FUNCTION CODE *	
10653	7F6A	.	.	*****	
10654	7F67	.	.	SNDCTB EQU \$-3	
10655	7F6A	30	37	DB 60Q,67Q ;OCTAL DIGITS	
10656	7F6C	93	93	DW DCNUM+B15 ;ACCUMULATE VALUE	
10657	7F6E	.	.	;	
10658	7F6E	41	41	DB 101Q,101Q ;<A>	
10659	7F70	B5	98	DW SNDCD1+B15 ;SEND ATTENTION CODE	
10660	7F72	.	.	;	
10661	7F72	46	46	DB 106Q,106Q ;<F>	
10662	7F74	FF	93	DW SNDCD2+B15 ;SEND FUNCTION CODE	
10663	7F76	.	.	;	
10664	7F76	20	20	DB 40Q,40Q ;SPACE	
10665	7F78	42	85	DW ESCAP1+B15	
10666	7F7A	00	7F	DB 0,177Q ;OTHER CHARACTERS	
10667	7F7C	00	80	DW B15 ;TERMINATE AND RESET	
10668	7F7E	.	.	;	
10669	7F7E	.	.	LI1 EQU \$	
10670	7F7E	8B	17	DW LDR3 ;A - ADDRESS	
10671	7F80	79	17	DW LDR0 ;B - IGNORE	
10672	7F82	C7	17	DW LDR10 ;CHECKSUM	
10673	7F84	9C	17	DW LDR4 ;DATA	
10674	7F86	AC	17	DW LDR060 ;E - EXECUTE LOADED CODE	
10675	7F88	00	00	DW BEGIN ;F - TERMINATE AND RESET	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 328
=====
10677    7F8A      . . .      ;*****
10678    7F8A      . . .      ; DFSTAB - DEFINE SOFT KEYS TABLE *
10679    7F8A      . . .      ;*****
10680    7F87      . . .      DFSTAB EQU  $-3
10681    7F8A      20 20 .      DB  40Q,40Q  ;SPACE
10682    7F8C      42 85 .      DW  ESCAP1+B15 ;IGNORE
10683    7F8E      30 39 .      DB  60Q,71Q  ;DIGITS <0>-<9>
10684    7F90      93 93 .      DW  DCNUM+B15 ;ACCUMULATE NUMERICAL VALUE
10685    7F92      41 41 .      DB  101Q,101Q ;<A> - ATTRIBUTE PARAMETER
10686    7F94      76 A2 .      DW  DFS100+B15 ;STORE DEFINED ATTRIBUTE
10687    7F96      4B 4C .      DB  113Q,114Q ;<K> - <L>
10688    7F98      AA 7F .      DW  DFT010   ;USE INDEX TABLE
10689    7F9A      . . .      ;*****
10690    7F9A      45 45 .      DB  105Q,105Q ;CAP E
10691    7F9C      86 A3 .      DW  EXSFKY+B15 ;EXECUTE SOFT KEY
10692    7F9E      . . .      ;*****
10693    7F9E      . . .      ;
10694    7F9E      . . .      ; LOWER CASE RANGE
10695    7F9E      . . .      ;
10696    7F9E      61 61 .      DB  141Q,141Q ;<A> - ATTRIBUTE PARAMETER
10697    7FA0      76 A2 .      DW  DFS100+B15 ;STORE DEFINED ATTRIBUTE
10698    7FA2      6B 6C .      DB  153Q,154Q ;<K> - <L>
10699    7FA4      AA 7F .      DW  DFT010   ;USE INDEX TABLE
10700    7FA6      00 7F .      DB  0Q,177Q  ;ALL OTHER CODES
10701    7FA8      48 85 .      DW  ESCEND+B15 ;ABORT ESCAPE SEQUENCE
10702    7FAA      . . .      ;
10703    7FAA      . . .      DFT010 EQU  $
10704    7FAA      7E 22 .      DW  DFS110   ;DEFINE KEY NUMBER
10705    7FAC      86 22 .      DW  DFS120   ;DEFINE LENGTH OF INPUT DATA
10706    7FAE      . . .      ;*****
10707    7FAE      . . .      ; ACCUMULATE SOFT KEY DATA TABLE *
10708    7FAE      . . .      ;*****
10709    7FAB      . . .      DFSTB2 EQU  $-3
10710    7FAE      00 7F .      DB  0Q,177Q  ;ALL CODES
10711    7FB0      32 A3 .      DW  DFS300+B15 ;ADD TO DATA LINE
=====
```


13255

13255/90010

2648A MICROCODE LISTING 'PT91'

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
A          0041      396
A2OUTB    1504      5859, 5854, 7810, 7829, 9781, 9785, 9817, 9821, 9825, 9875,
                9879, 9881, 9897, 9961, 9964, 9968
ABCKSL     005C      411, 4299
ABLNK      0020      378, 2963, 3189, 3216, 3282, 3394, 5382, 6126, 7385, 7400,
                7480, 8039, 9967, 10119
ABSTAK     FF5F      828, 832
ACINHB     0040      345
ADEL       007F      426, 2191, 6261, 9846
ALCC       0063      417, 5450
ALPHA      00C5      437, 4888, 7468, 7494, 7689, 8635, 9531, 9534, 9542, 9850
ALPHNM     00C7      439
ALPIN      0040      935, 1254
ALTIO      0010      853
ALTORG     9200      283, 284
ALTOUT     482A      213, 10270
AMPSND     0026      379, 1598, 5440, 9778, 9871
ANL        0080      845
ANR        0040      844
APIP       0002      355, 2407, 8884
ARPARN     0029      381, 1603, 9808
ATB010     1624      6129, 6132
ATBLEN     000E      6133, 1362, 1370
ATBLIN     161E      6125, 6133, 1349
ATBLOC     0008      6132, 5961, 9884
ATSIGN     0040      395, 6403
AUTOLF     0004      132, 1643, 4339, 5822, 5990, 6721, 9393
AUTTRM     0001      63, 5690, 6851
AVINHB     0020      344, 4842
B15        8000      447, 10355, 10357, 10359, 10365, 10371, 10375, 10380, 10382, 10386,
                10414, 10416, 10418, 10421, 10423, 10430, 10434, 10438, 10442, 10454,
                10459, 10461, 10463, 10465, 10469, 10553, 10556, 10559, 10561, 10578,
                10584, 10587, 10593, 10595, 10597, 10600, 10603, 10606, 10609, 10612,
                10615, 10618, 10620, 10627, 10629, 10631, 10633, 10646, 10648, 10650,
                10656, 10659, 10662, 10665, 10667, 10682, 10684, 10686, 10691, 10697,
                10701, 10711
B1LEN      FF38      897, 898
B1STAT     FF3A      895, 896
B1TYPE     FF39      896, 897
B2D050     0945      3040, 3010
B2D100     0958      3065, 3037, 3039, 3042, 3044
B2D110     095B      3068, 3071
B2D200     096A      3086, 3000, 3030
B2D210     0975      3093, 3088
B2D220     0976      3095, 3090
B2D8FL     003D      879, 7878, 10276, 10276
B2DBUF     FF3D      878, 879, 880
B2DDE      0934      3032, 1110
B2DEND     FF3B      881, 895, 5860, 7876, 10277
B2DPTR     FF3C      880, 881, 9928
B2LEN      FF35      900, 904
B2OUTB     1503      5855, 9847, 9849, 9895
B2STAT     FF37      898, 899

```


13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'PT91'

SYMBOL VALUE REFERENCED ON

```

=====
B2TYPE  FF36      899, 900
BACKT0  195B     6998, 7161
BACKT1  195E     7000, 6888
BACKT5  1A41     7139, 3429
BASE    FF00     591, 879, 1708, 2631, 2901, 2907, 2913, 3623, 4077, 5239,
              5308, 5319, 5328, 6535, 6758, 6769, 7076, 7164, 7200, 7213,
              7365, 7414, 7647, 7738, 8321, 8761, 8867, 8964, 9583, 9585,
              9786, 9800,10213
BASE2   FE00     593, 1370
BASEH   00FF     590, 591, 592, 1851, 2873
BASEH2  00FE     592, 593
BCKSPC  2177     8860,10393
BEGIN   0000     999, 1009, 1017, 1025, 1032, 1040, 1049, 1057, 2695, 1001,
              10675
BEL     0007     348, 2151
BELLIM  0008     456, 1632
BFSPCE  CFFF     663, 1198,10319
BINOCT  0902     2962, 4769, 4771
BINXMT  0002     461
BKT010  1A7B     7182, 7175
BKT050  1A7F     7193, 7178
BKT060  1A91     7203, 7197
BKT100  1A92     7208, 7167
BKT110  1AA0     7224, 7254
BKT120  1AA2     7226, 7251
BKT130  1AA6     7232, 7249
BKT150  1AB4     7247, 7228
BKT210  1989     7017, 7067
BKT220  1992     7025, 7058
BKT230  1994     7027, 7012
BKT240  19C1     7049, 7046
BKT250  19C4     7051, 7079
BKT300  19D3     7064, 7016
BKT310  19DF     7074, 7037
BKT400  19E9     7084, 7033
BKT410  1A05     7100, 7116
BKT420  1A0E     7105, 7112
BKT430  1A2A     7121, 7107
BKT450  1A34     7133, 7096, 7117
BKT500  1A49     7147, 7040, 7048, 7056
BKT510  1A4F     7150, 7125
BKTAB   1A58     7159,10533
BLKFIL  FF91     660, 661, 2614, 2631, 3353, 7629
BLKMDE  0002     131, 1955, 2413, 4339, 5071, 5685
BLKSM   000F     457, 1284, 2227, 2233, 2256, 2273, 2313, 3193, 3223, 3328,
              3360, 3937, 6212, 7269, 7758, 7916, 7972, 7990, 8009, 9994
BLKSZ   0010     458, 1281, 1300, 1322, 2276, 3200, 3505
BLKTRG  0001     87, 1900
BLKTRM  5004     241, 242, 1881, 5806, 6018, 6715, 6886,10008
BN2DA   0923     3002, 1109
BN2DE0  091D     2998, 1098
BN2DE1  0923     3004, 5449, 9962
BN2DE2  0926     3006, 5472

```

13255

13255/90010

REV 04/17/78

2648A MICROCODE LISTING 'PT91'

SYMBOL VALUE REFERENCED ON

```

=====
BN2DEC 092E 3028, 1097, 4776
BN0010 090C 2970, 2981
BNRYGO 2829 983, 984, 1918
BOT 0020 812
BRKDC 136C 5513, 6082, 6959
BSYCHK 283A 992, 993, 6301
BUFBGN FF8D 665, 666, 1202, 2075, 3887, 10320
BUFB SY 0080 855
BUFEND FF8B 666, 670, 1199, 2074, 2078, 3889
BUFMSG 1051 4788, 1094, 2092
C 0043 397, 1608
CALTST FF75 696, 697, 9800, 10271
CAPSLK 0001 130, 4339
CAR010 1136 4961, 4957
CARRET 1127 4955, 10627
CDSPEN FF77 694, 695, 9435, 9509, 9582
CHAIN 1AC6 7271, 1527, 1929, 2241, 2769, 2778, 3743, 4026, 5964, 6204,
6216, 6329, 7054, 7760
CHAIN0 1AC1 7265, 7507, 7530, 8363
CHAIN1 1AC2 7267, 10124
CHAR FF88 672, 673, 1782, 1868, 5349, 5576, 6510, 9119
CHARIN FF9C 651, 652, 1496, 1566, 1581, 1639, 5984, 5986, 9386, 9389,
9734
CHD000 1863 7418, 7351
CHD010 1AFD 7348, 7344
CHD020 1B29 7384, 7406
CHD050 1B5A 7410, 7379, 7401
CHD100 1B72 7439, 7424
CHD110 1B7B 7449, 7469, 7500
CHD120 1BA2 7474, 7454, 7463
CHD130 1BB0 7486, 7477
CHD140 1BB7 7492, 7490
CHD150 1BC5 7498, 7495
CHD200 1BC9 7506, 7461
CHD210 1BCD 7513, 7451
CHD250 1BE1 7529, 7459
CHD260 1BE5 7532, 7518, 7522
CHD400 1BEA 7544, 7462, 7496, 7514
CHD500 1BF3 7560, 7421
CHD510 1BFA 7567, 7572, 7581
CHD515 1C07 7573, 7569
CHD520 1C09 7579, 7571
CHEKCC 0040 80
CHI000 03D4 1678, 1670
CHI010 03F1 1700, 1772
CHI020 042B 1739, 1717
CHI030 0431 1744, 1742
CHI050 0446 1765, 1694
CHI100 0457 1779, 1690, 1713, 1719, 1724, 1767, 1770
CHI110 0462 1791, 1801, 1809
CHI200 048E 1847, 1835
CHI270 04B3 1867, 1859, 1863
CHINT 03E2 1686, 1674, 1682, 2177, 2444, 5985

```

13255

2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
CHINT0 03C2 1666, 1084
CHINT1 0457 1778, 1680, 2439, 2442
CHINT2 043E 1757, 2182
CHK010 1150 5000, 5042
CHK050 1153 5008, 4994
CHK060 115F 5018, 5035
CHK070 1162 5022, 4996, 5036
CHK100 1163 5027
CHK150 1165 5033, 5013
CHK160 116C 5040, 5012
CHKCT1 04CD 1891, 5711, 6687
CHKCTL 048B 1880, 1121, 10386
CHKFMO 1ACB 7291, 7160, 8759
CHKFMS 1ACF 7294, 1620, 2438, 2662, 2791, 3415, 3497, 3674, 3820, 4885,
5151, 5715, 5742, 5815, 6634, 7338, 7684, 7852, 7938, 8298,
8645, 8888, 8973, 9572, 9625, 9721, 10052, 10107, 10138
CHKFMT 1AD4 7298, 5782, 6755, 6766, 6870, 7423, 7547, 8264, 10203, 10243
CHKLIO 113C 4987, 5347, 9114
CHKLIM 1142 4990, 1068
CHKMLK 1ADA 7313, 2463, 6872
CHKRTN FF86 673, 674, 3679, 8641, 8672, 8925, 8929, 9726
CHKSFK 1AE5 7329, 1107, 1547, 1553, 1959, 1998, 2410, 2504, 4193, 4213,
4243, 4265, 4282, 4956, 5265, 5687, 5795, 6047, 6067, 7343,
7951, 8156, 8199, 9127, 9218, 9282, 9408, 9858, 10221
CHKSUM 0981 3119, 4473, 4519, 4637
CHRDEL 1AF2 7342, 7354, 10503
CHRD1 1B72 7438, 7393, 7603, 7823
CHRD2 1C11 7600, 5229, 8052, 8685, 9519, 10098
CHRINS 1C1A 7626, 7885
CHRLOC 0002 6134, 1368
CHRSET FF72 701, 702, 1440, 4180, 4267
CHRSTB 7F0F 10582, 4168
CHSAV FF98 655, 657, 7350, 7352, 7386, 7399, 7404, 7443, 7478, 7520,
7545, 7821, 7824
CIL 0001 783
CIR 0002 782
CKBRKY 000A 226, 6953
CKDSPF 1172 5049, 1766, 1858, 2189, 2399, 4197, 8974
CKEDIT 1178 5056, 2469, 2541, 3648, 3701, 4416, 5870, 5900, 8180
CKIOKY 0008 224, 4418, 5676
CKLNMD 118A 5080, 5731
CKPROT 1190 5089, 2671, 3355, 7644, 8330, 8959, 9474, 9642, 9667, 9678,
10251
CKRMTE 1195 5096, 1111, 5532, 5613, 6695, 6932
CLA010 1231 5253, 5257
CLBLXF 119B 5116, 1069, 1113, 4298, 5439, 5835, 6007, 10297
CLCMFL 1520 5890, 4196, 6513, 8845
CLEAR 099D 3149, 10492
CLEARL 1095 7935, 1077, 3433, 5153, 9173, 10498
CLEARS 11C0 5148, 1078, 10497
CLER01 10F5 8029, 5212
CLER02 10F6 8031, 7466, 7710
CLERL0 10B7 7965, 7837

```

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'PT91'
SYMBOL VALUE REFERENCED ON

```

=====
CLERL1  1DBA  7967, 7524, 7534
CLERLA  1DAD  7950, 7939, 9149
CLL120  1DD8  7995, 7999
CLL160  1DE3  8005, 7992
CLL310  1DF1  8015, 8010
CLL400  1DF3  8022, 7940
CLL510  1DF8  8034, 8042, 8053, 8061, 8066, 8076
CLL540  1E09  8046, 8038
CLL544  1E0C  8051, 8065
CLL550  1E12  8058, 8047
CLL580  1E23  8071, 8036
CLRAL1  1230  5251, 1192, 1982, 4506, 4624, 9062
CLRALL  122C  5238,10477
CLRDFL  1753  6466, 1499, 5150, 6269, 6867, 8194,10012
CLRMF2  0566  2015, 5307, 5432, 5912, 8982
CLRTRG  0000   265, 5140
CLRTRM  0002    64,10092
CLRXXN  11B9  5139, 5483, 6662
CLS100  11FA  5190, 5152
CLS110  1205  5201, 5194
CLS120  120A  5205, 5225
CLS130  120E  5211, 5196
CLS200  1215  5219, 5223
CLS210  1216  5221, 5230
CLSKFL  2403  9372, 9065, 9324, 9344
CMBASE  00FF   148, 149
CMDEXC  0008   810
CMFLGS  FFF8   157, 158, 1135, 1500, 1899, 2123, 4835, 5097, 5679, 5891,
      5925, 7843, 9185
CMND    FF55   836, 857, 2913, 6303, 6551
CMPLIM  FF46   877, 878
CMSTOR  FF00   149
CNTFAD  FFCE   598, 1849, 3005, 3034, 4319,10314
CNTRL0  FF62   816, 826
CNTXFR  0002   729
COMMA   002C   383
COMMON  FFFF   147, 148, 151
CONDIS  0001    34, 2398
CONDLF  0B69  3610, 4667, 4694
CONDTN  2814   973, 974, 6119
COUNT  FF84   678, 679, 3202, 3236
CR       000D   371, 1564, 1640, 1679, 1769, 2153, 2172, 5792, 5987, 6139,
      6718, 9390
CRA010  21E9  8952, 8956
CRA040  21B4  8921, 8896
CRA060  21C5  8931, 8907
CRA070  21D5  8938, 8935
CRA100  2203  8970, 8954
CRADV   21E3  8949, 8887
CRADV1  21AF  8911, 1104, 1441, 1783, 2609, 4740, 5836, 6073, 8932
CRAFLG  FF67   752, 774, 1687, 8913, 8964
CRI100  1C28   7643, 9640
CRI104  1C2E   7646, 9497, 9543, 9548, 9596

```

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'PT91'

SYMBOL VALUE REFERENCED ON

```

=====
CRI110  1C32  7650, 7674, 7683, 7685, 7690
CRI120  1C38  7655, 7653
CRI140  1C4A  7671, 7692, 7761, 7797
CRI150  1C7B  7697, 7688
CRI152  1C80  7701, 7703
CRI154  1C88  7706, 7700, 7708
CRI158  1C92  7716, 7678
CRI159  1C9D  7723, 7680
CRI160  1CA0  7731, 7719
CRI170  1CA1  7733, 7722
CRI180  1CA2  7735, 7773
CRI200  1C86  7753, 7676
CRI240  1CC7  7765, 7759
CRI260  1CDC  7784, 7775
CRI300  1CE2  7794, 7666
CRI305  1CEA  7799, 7662
CRI310  1D08  7817, 7831
CRI320  1D1C  7828, 7809
CRI330  1D31  7842, 7812
CRI400  1D65  7875, 7863, 7867, 7887
CRI450  1D7E  7891, 7870, 7879
CRI500  1D87  7913, 7702, 7707
CRI510  1D92  7920, 7917
CRLF    2222  8987, 4685, 4732, 4733, 4942, 4948, 4962, 6891, 8766, 8796,
      8817, 8975, 10026
CRP025  12AE  5346, 5321, 5330
CRP050  12B1  5348, 5337
CRP200  12C3  5364, 5359
CRP500  12D7  5388, 5365
CRPTAB  7F17  10591, 5310
CRRET   2366  9257, 1105, 4666, 8988
CRRET1  237F  9271, 5721, 5744, 8408, 10137
CRS100  1346  5470, 5459
CRSNGO  1315  5437, 1937
CRTOFF  0080  492, 1337, 3153, 4503, 6554
CSU100  0984  3123, 3127, 3131
CSU110  099A  3141, 3138
CTBDLY  0020  860, 2910
CTBLNK  FF53  858, 859
CTBLTM  FF52  859, 861, 2907
CTDCDP  282C  984, 988, 6295
CTIADR  FF33  904, 905
CTIBPT  FF2F  906, 907
CTICNT  FF2C  907, 908
CTICSM  FF2A  909, 910
CTIJMP  FFE0  172, 173, 1244
CTINTR  283D  993, 6396
CTISPT  FF31  905, 906
CTISTA  FF29  910, 914
CTITRL  FF28  908, 909
CTIVEC  FFE1  171, 172, 1242
CTLLIM  0020  377, 1616, 1693
CTLRED  2808  969, 970, 6120

```

13255
 2648A MICROCODE LISTING 'PT91'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
CTMON      282F      988, 989, 6318
CTRDKY     00A0      6094
CTSTAT     FF66      774, 785
CTUIN      0080      934, 1261
CUR100     130F      5433, 5426
CURAD2     2186      8878, 4676, 4711
CURADR     FFC3      615, 617, 1710, 1746, 2371, 2679, 2842, 2861, 3516, 4132,
                4895, 7142, 8411, 8463, 8922, 9145
CURADV     2189      8881, 1749, 4730, 8879, 9713
CURCOL     FFC1      623, 624, 1629, 1759, 2622, 2624, 2655, 2704, 2745, 3245,
                3370, 5304, 5319, 5374, 5448, 6148, 6609, 6758, 6768, 7093,
                7162, 7358, 7412, 7738, 7746, 7847, 7883, 8321, 8761, 8867,
                8894, 8951, 9013, 9141, 9147, 9172, 9272, 9463, 9481,10003,
                10093,10110,10205
CURFKY     FFA4      639, 640, 5965, 6255, 6260, 9355, 9357
CURPD      2246      9023,10489
CURPH      1E9D      8192, 1073, 1384,10532
CURPH1     1EA2      8197, 6981, 8830, 8944, 9653
CURPHD     1238      5264, 1074, 7018,10493
CURPL      2233      9005,10491
CURPL1     223B      9012, 9001
CURPL2     2251      9035, 9016
CURP01     1280      5316,10600,10609
CURP02     128B      5325,10606,10615
CURP03     1296      5334,10603,10612
CURP04     12C9      5373, 7166, 7202, 7241, 8799
CURPOS     126B      5303,10564
CURPR      2228      8994,10490
CURPRT     237C      9269, 3434, 3529, 3819, 5267, 8198, 9150, 9266,10494
CURPU      2253      9040,10488
CURPU1     225B      9047, 9030
CURROW     FFC0      624, 625, 1338, 1725, 2525, 2530, 2748, 2863, 3286, 3385,
                3532, 3623, 3698, 3818, 4050, 4058, 4869, 5296, 5328, 5342,
                5360, 5395, 5457, 6501, 6898, 6901, 7004, 7013, 7020, 7035,
                7069, 7070, 7108, 7113, 7122, 7134, 7152, 7173, 7345, 7387,
                7394, 7855, 7892, 7944, 8270, 8279, 8291, 8421, 8456, 8457,
                9048, 9132, 9151, 9864, 9876,10069,10141,10225
CURSEN     130A      5430,10525
D          0044      398, 8466
DATATR     0040      824
DATCOM     0020      854
DBLHOL     0010      811
DC2        0012      374
DC2GO      1359      5490, 1933
DC2SND     0080      53, 5709, 6684
DC3        0013      375, 1681
DCC010    137D      5538, 5533
DCERR      1382      5543, 6550
DCH010    25AC      9735, 9744
DCH020    25AF      9737, 9740
DCH100    25C2      9751, 9718
DCHAR     FF89      671, 672, 1702, 3248, 3280, 3308, 3373, 4673, 4887, 5381,
                7627, 7647, 7681, 7736, 7795, 9416, 9433, 9455, 9490, 9555,
  
```

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
                                9585, 9647, 9700, 9725
DCIOFF      0010      111
DCJMP0      0080      68
DCJMP1      0001      72
DCJMP2      0002      73
DCJMP3      0004      74
DCJMP4      0008      75
DCJMS2      5006      243
DCJMSK      5005      242, 243
DCM010      13BE      5601, 5595
DCMCT1      137E      5540, 1155, 5914
DCMCTL      1373      5530, 1893, 5141, 5485, 5496, 5515, 5669, 5880, 6029, 6821
DCMERR      0001      96, 4305
DCMINT      1365      5504, 1031
DCMNUS      13BC      5599, 1066, 10595
DCN005      139D      5575, 5573
DCN010      13AC      5584, 5587
DCNUM       1393      5569, 1064, 10597, 10656, 10684
DCPLUS      13B7      5593, 1065, 10593
DCTEST      13CA      5612, 1095, 5678, 10540
DCXB2D      13D7      5628, 1106, 4245, 4958, 6049, 6069, 8563, 8933, 8942, 9238,
                                9245, 9645, 9723, 9754
DECRDX      000A      138, 1975
DEFKEY      0001      954, 9064, 9116, 9288
DEFSKY      0008      90, 4195, 4215, 4836, 9186
DELAY       13E4      5644, 1101, 1646, 9396
DELAY0      13DD      5636, 10487
DELTRM      0000      740, 2857, 8537
DELWRP      1AEA      7337, 10502
DENTAB      7F0B      10576, 9410
DEVFLG      FE7F      932, 939, 1265, 2928, 6394
DFCTOF      15E1      6057, 6083
DFLGS       FF6E      726, 737, 1150, 1618, 1884, 1951, 4431, 4779, 5629, 6008,
                                6453, 6467, 6739, 6808, 8979, 9329, 10020
DFS100      2276      9076, 10686, 10697
DFS110      227E      9083, 10704
DFS120      2286      9092, 10705
DFS200      2288      9103
DFS210      2298      9111, 9108
DFS220      229B      9113, 9079, 9087, 9110
DFS230      22FC      9160, 9156, 9158
DFS250      2308      9170, 9144
DFS300      2332      9217, 10711
DFS350      2359      9237, 10355
DFS360      2360      9244, 10357
DFSFKY      2264      9059, 10569
DFSTAB      7F87      10680, 9067
DFSTB0      7DF8      10363, 2000
DFSTB2      7FAB      10709, 9163
DFSTB3      7DEF      10353, 9232
DFT010      7FAA      10703, 10688, 10699
DIS020      2468      9476, 9574
DIS030      2476      9487, 9475, 9504, 9520, 9536, 9546, 9571, 9573

```

13255
 2648A MICROCODE LISTING 'PT91'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
DIS035  24A2  9516, 9492
DIS040  24AA  9525, 9502
DIS042  248F  9541, 9533
DIS043  24CE  9552, 9535, 9643
DIS044  24D8  9559, 9512
DIS045  24DB  9568, 9530
DIS050  24EA  9579, 9495, 9501
DIS054  24F8  9588, 9581
DIS057  2503  9598, 9594
DIS060  2508  9606, 9457
DIS070  2510  9624, 9610
DIS080  2527  9633, 9608
DIS090  2537  9641, 9619, 9637
DIS092  253D  9644, 7645, 9480, 9668, 9679
DIS093  254E  9652, 9628
DIS100  2555  9664, 9468, 9612
DIS110  256A  9677, 9600, 9618, 9666
DIS114  2573  9681, 9675
DIS120  09CF  3198, 3201
DIS140  09E1  3215, 3229
DIS160  09FA  3235, 3204
DIS170  0A15  3255, 3247, 3261
DIS175  0A16  3257, 3250
DIS180  0A1D  3265, 3318, 3332
DIS210  0A33  3283, 3281
DIS220  0A4A  3304, 3181
DIS240  0A54  3315, 3220
DIS400  0A5A  3323, 3188
DISCNT  0006   271, 5520
DISLN1  10D3  4868, 1745, 2181, 3297, 3399, 5800, 8174, 8467,10147
DISLN2  10D6  4870, 5638
DISLN3  10D9  4874, 1857
DISLN4  10DA  4876, 5555, 6548
DISLNK  10CA  4859, 3475, 3594, 8258
DISMDM  1371  5519,10530
DISPC0  2441  9418, 1103, 4705, 4710
DISPC1  2443  9430, 4277, 4889, 6638
DISPC2  2445  9432, 4936, 6798, 7403
DISPEN  2432  9407,10567
DISPLO  250E  9609, 5383
DISPL1  09AB  3177, 7744, 9680
DISPL2  09B7  3184, 9674
DISPLA  244C  9454, 7631, 9651, 9654, 9686
DISPLC  243C  9415,10578
DISPLY  0004   851
DISPST  FFFE   151, 152, 2370, 2855, 4846, 8116, 8173
DLY010  13ED  5650, 5659
DLY020  13F1  5653, 5657
DMAOFF  0060   491, 4860
DOOCTI  2835   990, 991, 1241
DOWN    0001   361, 9025
DPS100  1439  5699
DPS200  143F  5707, 5686
  
```


13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
DPS210 1449 5712
DPS215 144C 5714, 5732
DPS220 1457 5726, 5688, 5691
DPSEN1 1466 5736, 6900
DPSEND 1409 5675, 6081
DPSGO 1476 5756, 1939
DSG010 1481 5761, 5786, 5801
DSG020 1489 5768, 5772
DSG100 1498 5777, 5770
DSG110 14B1 5791, 5783
DSG200 14C8 5805, 5758, 5778
DSG210 14D7 5814, 5781
DSG220 14EB 5827, 5809, 5816, 5819
DSG225 14EE 5829, 5810
DSG230 14F5 5833, 5773
DSM010 1E3F 8106, 8114
DSM500 1E57 8124, 8096
DSM510 1E5D 8127, 8136
DSP010 04E0 1906, 1912
DSP020 04FD 1928, 1908
DSPASC 2508 9605, 9142, 9716
DSPBGN FFAA 636, 637, 1210, 1290, 2088, 3892, 4323
DSPBTM 0040 801,10215
DSPCHO 258F 9714, 4963
DSPCH1 25C3 9753, 6760, 6771
DSPCHR 2583 9706, 1108, 4244, 4247, 4677, 4959, 6072,10371
DSPEND FFA8 637, 638, 1207, 1271, 1276, 2087, 2090
DSPFNC 0001 119, 5051, 6039, 6061
DSPLIM F7FF 575, 1189, 1206
DSPMG2 1E39 8100, 1120
DSPMS0 1E2F 8091, 5549, 6503
DSPMS1 1E30 8093, 1399, 4430, 4433, 5617
DSPMSG 1E33 8095, 1062
DSPSTR FE4F 583, 1348, 1362, 1368, 1370, 8101, 8115
DSPTAB 0501 1932, 1942, 1903
DSPTCH 04D2 1898, 1489
DSPTST 257D 9699, 4699, 4719, 4918, 9161
DSPTYP FFAE 629, 635, 1383, 7295, 7330, 9190
ECONTF FFC0 597, 598, 603, 627, 1130, 1170, 1401, 1855, 3048, 3079,
3965
ECOUTB 14FE 5852, 9779, 9809, 9833, 9837, 9840, 9852, 9873
EDIT 0010 123, 5058
EDTWRP 0008 66
EI1 7E78 10474,10440
EI1A 7E82 10481,10445
EI2 7E88 10486,10448
EI3 7EE4 10539,10451
ELM100 0A92 3381, 3368, 3372, 3375
ELM110 0A9E 3393, 3397
ELM130 0AAF 3404, 3402
ENDBLK 0007 272, 5484
ENDCOL FF21 924, 925
ENDDSP 0080 802,10215

```

13255
 2648A MICROCODE LISTING 'PT91'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
ENDPR      00C1      433, 2690, 5224, 6126, 6130, 6138, 6139, 6603, 6999, 6999,
              7948, 8564, 8564, 8934, 9478, 9835
ENDROW     FF20      925,10073,10145,10259
ENDTST     0006      222, 2898, 4737
ENHLIM     00BF      431, 7453, 7673
ENHNCF     00FF      6110, 1583
ENHOUT     FF76      695, 696, 9773, 9786,10136,10273
ENL100     1519      5878, 5871
ENR100     1534      5908, 5901
ENTLCL     150B      5869, 2199
ENTRCD     0098      6091, 5681
ENTREM     1526      5899, 2126
ENTREN     1822      6646, 1118,10528
EOF        0001      818
EOL        00CC      443, 1718, 3244, 3316, 3330, 3366, 4400, 4931, 5291, 6126,
              6128, 6138, 6139, 7441, 7458, 7533, 7677, 7786, 7954, 8629,
              8715, 8715, 8744, 9493,10047
EOLADR     FF94      658, 659, 3186, 3285, 7785
EOLMOV     0A71      3358, 3179
EOLMV      FF90      661, 662, 3407, 8602, 9638
EOLMV0     0A69      3352, 2859
EOP        00CE      444, 2325, 2493, 3803, 4772, 4789, 4806, 4809, 4812, 4815,
              4928, 5163, 5213, 5226, 6426, 8073, 8367, 8631, 8746,10043
ERREOP     107A      4805, 4763
ERRFLG     FFF7      158, 159, 4303, 4366, 4715, 6563
ESC        001B      376, 1574, 1771, 2155, 5853, 6054, 6126, 6925
ESC010     0526      1958, 1953
ESCAP0     0534      1976, 1961, 4169, 6517, 6813, 9411
ESCAP1     0542      1983, 5589, 5607, 9231, 9246,10359,10618,10646,10648,10650,
              10665,10682
ESCAPA     0532      1974, 5311, 6585, 9068, 9164, 9233
ESCAPB     053A      1979, 5351, 9121
ESCAPE     0511      1947, 1102,10375
ESCEN1     055D      2001, 1995, 1999, 6042
ESCEND     0548      1991, 1067, 1438, 1865, 2190, 5605, 6060, 9289, 9294, 9297,
              9309,10414,10416,10418,10421,10423,10469,10483,10513,10515,
              10561,10568,10587,10620,10701
ESCFLG     FFD1      184, 187, 1626, 1860, 1984, 2004
ESCINP     0008      720, 1460, 1956, 2005
ESCLWD     00E4      6099, 1597
ESCOUT     191C      6924, 4300, 5441,10299
ESCSO      008E      6098, 1607
ESCTAB     7E3D      10427, 1960
EVD        0002      819
EW         0080      814
EXK010     23B1      9308, 9325
EXK015     23C3      9323, 9340
EXK020     23C9      9326, 9321
EXK030     23D1      9333, 9339
EXK040     23E5      9342, 9331, 9335
EXK050     23EA      9348, 9304
EXK060     23EF      9352, 9354
EXP010     25F0      9791, 9777
  
```

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
EXP020 2614 9819, 9802
EXP030 2619 9823, 9799
EXP100 261E 9830, 9772
EXP110 2634 9844, 9834
EXPAND 25CA 9768, 1088, 10058
EXSFKY 2386 9281, 10691
EXSK1 2388 9313, 1561
EXTB2D 0001 799
F 0046 399, 4362
FOCODE 00EF 6101, 6105, 1535, 1539, 9298
F1CODE 00F0 6103
FCR005 2054 8606, 8616
FCR010 2058 8612, 8623, 8638, 8646, 8654, 8674, 8684, 8686
FCR100 206C 8628, 8621
FCR110 20A2 8655, 8652
FCR150 20A4 8661, 8636
FCR160 208C 8671, 8664, 8667, 8669
FCR200 20C3 8680, 8634
FCR250 20D3 8690, 8683
FCR260 20D4 8692, 8614
FCR400 2046 8574, 8536
FCT200 1576 5980, 5969, 5988, 5991
FCT210 157A 5983, 5993
FCTAD1 01DE 6105, 6107, 6107
FCTADJ FFDD 6107, 5954
FCTK2D 0010 732, 1619, 5970, 6009, 6268, 9330
FCTKEY 154A 5951, 9318
FDESC 15EE 6066, 10629
FDESC1 15FA 6071, 4248, 6052, 6068, 9224
FDISOF 15CC 6046, 10631
FDISON 15BF 6036, 10512
FDISTB 7F43 10625, 6041
FDO100 15C1 6038, 6032
FF 000C 370
FILCHR FF8F 662, 665, 2220, 2277, 7968, 8000
FILL 00C3 435, 1723, 2217, 3260, 3376, 7460, 7523, 7679, 7836, 7997,
      8060, 9848, 10049
FILNUM FF5E 832, 833
FILRED 0004 789, 1473
FIVE 0035 391, 1038
FKEYGO 1595 6005, 1938
FKG010 15A1 6014, 6022
FLDSEP 00C4 442
FLDSR 1F3A 8324, 3821, 5202, 8828, 8940, 9650, 9900, 10178
FLDSR1 1F2F 8318, 8284, 10077, 10255
FLDSR2 1FFA 8489, 8075, 8386, 8501, 10162
FLDSRB 1FF9 8486, 8906
FLDSRX 1F96 8395, 4226, 6799, 10031
FLINE FF9F 642, 649, 2374, 2539, 2550, 3466, 3604, 8241, 8244, 8295
FLS010 20DF 8722, 8735
FLS020 20EA 8728, 8725
FLS030 20EB 8730, 8727
FLS035 20F0 8734, 8747

```

13255
 2648A MICROCODE LISTING 'PT91'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
FLS040  20F3  8736, 8745
FLS050  20F6  8743, 8732
FMTCTL  FF8A   670, 671, 8904, 8972
FNCLIM  00A1  6096, 1520
FNCLWR  0098  6095, 1522
FNCTAB  1600  6080, 1525
FNDCH   201E  8530, 2691, 8939, 9479
FNDCH0  201B  8528, 2677, 8927
FNDCHR  204D  8600, 2856, 8575
FNDCHU  2037  8562, 8352
FNDCU1  2042  8570, 8338, 8565
FNDLS0  20D6  8713, 7866, 9136, 9955
FNDLST  20DA  8717, 7032
FNDRAM  056C  2032, 1201, 1209, 2045
FNDTAB  163B  6147, 6277, 6285
FNDTB1  163F  6150, 7217, 8772
FNDTB2  164C  6170, 1079
FOF010  15D8  6051, 6048
FORGN   0080   126, 4399
FORMAT  0008   122, 6182, 6871, 7300, 8184, 8265
FORMOF  1654  6181,10511
FORMON  1E8E  8179,10510
FOUR    0034   390, 1030
FPS     0004   809
FRBLKS  FFAC   635, 636, 1298, 2221, 2242, 2586, 2588, 5165, 5174, 6194,
        6873
FRC010  1661  6195, 6213
FRC050  1670  6207, 6217
FRC100  1682  6221, 6201
FRCPTY  0080   82
FRCRST  0004   89, 1136, 3151, 5551, 6507, 6512
FRECNT  165C  6192, 1075, 3649, 6223
FRM010  0580  2049, 2037, 2040
FRNCT1  168A  6230,10482
FRNCT2  168F  6236,10484
FRNMD1  000E   230, 6231
FRNMD2  000F   231, 6237
FRSALT  4829   212, 213, 1439, 4404
FRSOUT  0010   721, 9898, 9947
FRSTBL  FF92   659, 660
FS2000  1FF6  8484, 8505
FS2005  1FF9  8487, 8507
FSR080  1F4B  8336, 8389
FSR100  1F58  8351, 8323, 8331, 8380, 8390
FSR120  1F5E  8359, 8339
FSR140  1F96  8396, 8361, 8368
FSR200  1F9C  8405, 8353
FSR240  1FB8  8423, 8451
FSR300  1FCB  8444, 8448
FSR340  1FDA  8455, 8426
FSR360  1FF1  8465, 8420
FST     0004   840
FSTBIN  000A   275
  
```

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
FSTRAM  9100      21, 144, 588, 1174, 1175, 4530, 4600
FSTSND  0020      48
FTB100  164E     6172, 6175
FULDUP  0080      25, 2428
FWD     0002      839
GAP     0020      778
GBL100  0656     2250, 2234
GBL200  0664     2272, 2245
GBL210  066D     2278, 2282
GDC010  05C8     2130, 2152, 2156, 2158, 2178
GDC020  05D0     2138, 2192
GDC025  05ED     2159, 2154
GDC027  05F4     2165, 2146
GDC030  060A     2176, 2170, 2173
GDC050  0616     2187, 2132
GDC100  0624     2197, 2124
GDS010  2646     9857, 9932
GDS020  268F     9894, 9889, 9892
GDS030  2689     9944, 9870
GDS040  26D6     9959, 9957
GDS045  26EC     9974, 10023
GDS050  26ED     9977, 9859, 9949
GDS060  26F5     9982, 9995, 10050, 10067, 10167
GDS100  270F     10001, 9987
GDS110  2727     10019, 10011
GDS150  2735     10030, 9981, 10044, 10078, 10101
GDS160  2738     10033, 9862, 9868, 10151
GDS200  273C     10042, 9989
GDS210  2764     10064, 10053
GDS220  2776     10074, 10163
GDS225  2786     10081, 10076
GDS230  2789     10090, 10046
GDS300  279F     10106, 10048
GDS310  27BE     10123, 10108, 10109, 10113
GDS320  27C1     10129, 9996
GDS350  7D15     10176, 10140
GDS360  7D1B     10179, 10139
GEN     0020      843
GETADR  FF73      697, 701, 9902, 9952, 9978, 9985, 10080, 10095, 10118, 10134,
          10148, 10168, 10269
GETBUF  0587     2073, 1228, 1252
GETDC1  0610     2180, 1450, 9228
GETDCM  05B8     2120, 1090, 1455, 9400
GETDSP  26A3     9923, 1086, 5769, 9969, 10059
GFUNMX  0098      346, 1514
GINMOD  0010      347, 2420
GO      00F3     1126, 1004
GO010   0117     1144, 1140
GO1     0119     1149, 4259
GTB005  0596     2079, 2091
GTB010  0599     2086, 1274, 2077
GTB100  05AE     2097, 2076, 2089
GTBLK   062D     2219, 3190, 3217

```

13255

2648A MICROCODE LISTING 'PT91'

SYMBOL VALUE REFERENCED ON

13255/90010

REV 04/17/78

```
=====
GTBLKF 0628 2216, 2308, 3324, 3500
GTEXT 0002 356, 1704, 3617, 8755, 8863, 9260, 9709
GTF010 16A5 6267, 6258
GTFCTK 1694 6254, 5981, 6015, 9334, 9353
GTMOD1 117E 5065,10075
GTMODE 1184 5069, 1100, 5780, 5808, 5972, 6714
GTNWLN 0677 2305, 2369, 2802
H 0048 400
HANGU0 1385 5548, 1093, 1230, 2093, 2188, 4782, 5616, 6430, 6968
HANGUP 1388 5550, 1119
HDC100 123F 5271, 5284
HDC200 1258 5289, 5278
HDC210 1267 5295, 5293
HNDSHK 0040 50, 5709, 5710, 6670
HNG010 138D 5554, 5556
HOL 0010 779
HOLCNT FF51 861, 862
HP2648 0004 358, 1234
HRDER1 0010 822
HRDERR 0004 820
HTAB 2100 8752,10394,10496
HTB100 2123 8779, 8822
HTB120 2128 8782, 8809
HTB130 2144 8807, 8784
HTB140 2148 8814, 8781
HTB160 212C 8788
HTB200 2155 8827, 8760
HTBCLR 1682 6284,10476
HTBLEN 000A 689, 690, 5240
HTBSET 16AC 6276,10475
HTBTBL FF78 690, 694, 5239, 6155, 7194
HUP050 1EF3 8262, 8221
HUP060 1F08 8277, 8200
HUP100 1F19 8288, 8205
HUP110 1F1C 8290, 8266
ICH010 215E 8839, 8855
ICHOFF 2163 8843,10505
ICHON 215C 8837,10504
IDG055 7D50 10237,10283
IDG060 7D65 10247,10242
IDG070 7D69 10250,10245
IDG090 7D7A 10257,10252
IDG100 7D80 10264,10244
IDG110 7D9E 10280,10223
IGNTRM 0001 741, 2610, 2833, 6795, 7008, 8531,10238
INERMS 1069 4797, 6427
INI010 0143 1176, 1181
INI020 014D 1191, 1194
INI110 01B5 1255, 1251
INI130 01C6 1264, 1260
INI210 01F4 1313, 1328
INI220 01FB 1318, 1316
INI310 0228 1357, 1375
```

13255

13255/90010

2648A MICROCODE LISTING 'PT91'

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
INIT      0139      1168, 1133, 1137, 1143
INITD0    7D1F    10198, 1085
INITD1    7D97    10275, 7803, 9769, 9872, 9951
INITDG    7D2E    10206, 1117, 5757, 6843,10201,10204
INITDS    06AB    2368, 1334, 1389
INPDEV    FF4E      869, 870
INSCHR    0002      120, 7369, 8840, 8846, 8962, 9636
INSWRP    0002      88, 7845, 8844, 8852
INTERR    1729    6421, 6441
INTFLG    FFF6      159, 160, 2923, 2936, 5652, 5655
INTRPT    173D    6439, 1016, 1048, 1056
INTVEC    9165      144, 145, 1219, 2885, 5505, 6386, 6440
INVRS     0082      493, 6131, 6137
IOBASE    0080      471, 475, 486, 498, 507, 514
IOBNGO    16BA    6294,10529
IOBSYC    16C0    6300, 3150, 4255, 4434, 6307
IOBUF     FC00      578, 579, 580, 4505, 4536, 4623
IOBUF1    FC00      581
IOBUF2    FD00      582, 4766, 4774
IOBUFH    00FC      579, 580
IOBUFL    0000      580
IOCCNT    FFD5      890
IOCDEV    FFDB      885
IOCDPT    FF4C      871, 872
IOCERR    FF4F      866, 869, 4361, 6306
IOCINP    FFD9      887
IOCKEY    2802      967, 968, 6115
IOCMND    FFD7      889
IOCNTL    281A      978, 979, 6312
IOCOUT    FFDA      886
IOCRCL    8700      487, 1760, 6390, 9273,10007,10116
IOCRRW    8720      488, 3154, 4504, 6555, 8118
IOCSGN    FFDD      174, 175, 1980, 4988, 5570, 5602
IOCTCO    8B00      499, 2916, 6429
IOCTDI    8B20      502
IOCTDO    8B20      501
IOCTGO    16D2    6311,10556
IOCTMN    16D8    6317, 1488, 6952, 9401, 9738
IOCTSI    8B00      500
IOCTU     8B00      498, 499, 500, 501, 502
IOCTYP    FFD8      888
IODATA    FFDE      173, 174, 757, 4991, 5009, 5581, 5588, 5666, 6422, 6818,
          9104, 9112, 9291
IODISP    8700      486, 487, 488
IODNGO    2820      980, 981, 1940
IOERRB    0020      459, 4364
IOFLG2    FF64      797, 805, 3640, 5066,10054,10165,10213
IOFLGS    FF65      785, 797, 1472
IOI010    1723    6407, 6410
IOI020    1728    6411, 6405
IOINTR    16FF    6385, 1039
IOKB     8300      475, 476
IOKBC0    8380      476, 1732, 1853, 4878

```

13255

13255/90010

2648A MICROCODE LISTING 'PT91'

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
IOKEYS  16DE  6326, 5682, 6084, 6085, 6086, 6087, 6088, 6089
IOKYTB  1612  6114, 6327
IOORG   2800  966, 967, 1257
IOPSGN  FFDC   175, 176, 4989, 4993
IOPTR1  8D00   507, 508, 509, 510
IOPTR2  8500   514, 515, 516, 517, 518
IORDGO  2823   981, 982, 1916
IORMG1  16F5  6367, 1258, 4460, 6347
IORMGO  16E5  6344, 1096, 1160, 1249, 4421, 6296, 6302, 6313, 6319
IOSTA0  FF48   875, 876
IOSTA1  FF49   874, 875
IOSTA2  FF4A   873, 874
IOSTA3  FF4B   872, 873
IOSTGO  281D   979, 980, 1936
IWRPON  216D  8851,10501
JMP     00C3   448, 1131, 1243, 1400
KBDCSW  FFFC   153, 154, 2427
KBDLOK  0040   733, 6454, 6460, 6482
KBEN    1746  6452, 5135
KBEN1   174C  6457,10526
KBFCTK  FF71   702, 704
KBJMP2  FFFA   155, 156, 4427, 5689, 6850,10091
KBJMP3  FFF9   156, 157,10331
KBJMPR  FFFB   154, 155, 2397, 3611, 4330, 5081, 5708, 6669, 6683, 8976,
          9264
KBLOK   175E  6485, 6705
KBLOK0  1759  6481,10527
L       004C   401, 4899, 9157, 9891, 9963
LABEL   0080   357, 1704, 9709
LADDR   FF05   756, 6523, 6537, 6541, 6557, 7007, 7085, 7088, 7138, 7148,
          8412, 8462
LCHAR   FF69   750, 751, 1869, 6053
LCHKSM  FFD7   758, 6506, 6526, 6528, 6569, 7001, 7031
LCI050  0701  2437, 2400, 2411, 2415, 2418, 2422
LCN010  242B  9399, 9391, 9394
LDATA   FFDE   757, 6522, 6535, 6567, 7005, 7011, 7151
LDR0    1779  6509, 6529, 6577,10671
LDR035  1792  6525, 6542
LDR060  17AC  6547, 6553,10674
LDR10   17C7  6562,10672
LDR3    178B  6521,10670
LDR4    179C  6534,10673
LDRCHK  0004   98, 6565
LDRMSG  1074  4803, 6502
LDRTAB  7F53  10638, 6515
LEFT    0002   362, 9007
LF      000A  369, 1647, 1669, 2174, 5992, 6724, 9397
LFPOS   0010   42
LFTBKT  005B   410, 9836
LFTBRC  007B   424, 9832
LFTCTU  0001   849
LFTMGN  FF8F   625, 626, 6762, 6769, 7164, 7213, 7390, 7882, 9270
LI1     7F7E  10669,10640,10643

```


13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
LID050 0AD4 3432, 3421
LID200 0AF0 3472, 3459
LID300 0AF6 3481, 3467
LII200 0B61 3600, 3583
LINDEL 0AB7 3414,10500
LINDLO 0ADA 3448, 3422, 4030, 4100
LININO 0B27 3527, 3430
LININI 0B3C 3551, 3523
LININA 0B39 3547, 4034, 4109
LININS 0B00 3496, 7869,10499
LINWRP 0004 38, 3612, 8977
LLINE FFA1 641, 642, 1381, 2311, 2329, 2480, 2487, 2810, 5173, 5176
LNF100 0B8A 3632, 3626
LNFEED 0B6F 3614, 1087, 5798, 5823, 8989, 9239,10146,10177
LNKLIM 00D0 445, 3928, 7675, 9988
LNKSAV FF96 657, 658, 2999, 3029, 3096, 3100, 7754, 7801, 7834, 7918,
8370, 8459, 8793, 8801
LOADR 1763 6499,10565
LOADR1 176E 6504,10566
LOCKKB 0001 217, 6486
LOCLI0 06C4 2396, 1580
LOCLIN 06CF 2404, 1638, 9388
LOCLN2 240A 9385, 1116, 9336, 9398
LP 0040 813
LPM 0001 807
LSTCOL FFC8 609, 610, 1747, 3531, 4135, 7137, 8397, 8409
LSTDOD FFC6 613, 614, 3671, 8375, 8388, 8622, 9583,10272
LSTFMT FFC5 614, 615, 2307, 3673, 5206, 6625, 6640, 8508, 8643, 8905,
8971
LSTFWD 0002 808
LSTLIN FFC9 606, 609, 2373, 2471, 2752, 3266, 3418, 3520, 3522, 3666,
4083, 5155, 5275, 7006, 7029, 7050, 7057, 7087, 7090, 7140,
7149, 7857, 7966, 8461, 8902, 9882
LSTLU1 0BA8 3667, 2830
LSTLU2 0BA9 3669, 3528
LSTLUP 0BA5 3665, 2812, 4129
LSTRED FF25 915, 916
LSTROW FFC7 610, 613, 1339, 2749, 2872, 4037, 4077, 4111, 4155, 5272,
5282, 5292, 7019, 7135, 8271, 8280, 8293, 8458
LWBUF 00B0 664, 1200
LWDSP 00D0 577, 1208
MAXCOL 004F 454, 2376, 3246, 3371, 5317, 7026, 7196, 7660, 7721, 7822,
7833, 8182, 8343, 8398, 8406, 8571, 8764, 8790, 8955, 9017,
9036, 9093, 9135, 9137, 9148, 9617, 9954, 9956,10025,10112
MAXROW 0017 453, 3625, 3812, 3973, 4081, 5273, 5326, 5397, 5404, 5405,
5637, 6500, 7101, 8117, 8424, 9046, 9051
MAYEOL 0040 490, 1726, 3287, 3366
MAYEOP 0020 489, 4105
MDFLG1 FFF4 161, 162, 2168, 4398, 5050, 5057, 5182, 7299, 7314, 7365,
7414, 8961, 9220, 9226, 9635,10344
MDFLG2 FFF3 162, 163, 1642, 1954, 2121, 2412, 4338, 5070, 5684, 5821,
5873, 5903, 5989, 6720, 9392
MEMLOK 0004 121, 3692, 3706, 3770, 5183, 7316,10346
=====

```


13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
NULL      0000      368
NUM2K     0800     446, 4445, 4448, 4472
NUMBER    00C6     438, 8666
NUMSWP    000F     627, 628, 9195
NWRWST    0080     724, 5306, 5338, 5390
NXB060    0C70     3888, 3903
NXB100    0C7E     3895, 3886
NXB150    0C87     3900, 3898
NXB200    0C8C     3904, 3891, 3894
NXSBLK    0C67     3882, 4518, 4612, 4636
NXT040    0C3E     3816, 3977
NXT100    0C4E     3835, 3815
NXT110    0C53     3841, 3848
NXT120    0C62     3854, 3846
NXTCH0    0C8F     3923, 5290, 6257, 7493, 9146
NXTCHR    0C90     3925, 1089, 2661, 2682, 5222, 7450, 7497, 7568, 7587, 7805,
          7949, 8035, 8500, 8613, 8723, 9488,10267

NXTPG1    0C4E     3836, 5407
NXTRED    FF27      914, 915
NZEXIT    0C0A     3750, 2309, 2792, 3730, 8503, 8608,10256
OCTRDX    0008     139, 6516, 6812
OPSTOR    FFD0     589, 590, 597
OTHER     FF56     834, 836
OUTDEV    FF4D     870, 871, 1240
OVRFLO    7D00    10156,10154
P         0050     403, 6370
PAGSTR    0008     40, 5082
PARM1     FFDB     176, 177, 682, 885, 6608, 6628, 9094, 9140
PARM2     FFDA     177, 178, 683, 886, 9078, 9153
PARM3     FFD9     178, 179, 684, 887, 9085, 9129, 9139, 9171
PARM4     FFD8     179, 180, 888
PARM5     FFD7     180, 181, 758, 889, 7800, 7832
PARM6     FFD5     181, 182, 756, 890, 7807, 7818
PAROT1    0CA7     3959
PAROT2    0CA6     3957, 4326,10322
PAROT3    0CA5     3955
PAROT4    0CA4     3953, 4334, 4355,10335,10339
PAROUT    0CA8     3961, 4332, 4341, 4368, 4384,10327,10333,10337,10347
PERIOD    002E     385
PLUS      002B     382
POLL      0040     113, 6388, 6392
PRCCTL    FFF5     160, 161, 1128, 1217, 2886, 5647, 6387
PREND     17F0     6602,10514
PREVPG    0C82     3972,10509
PRINTR    0008     852
PRM010    7F00    10563,10550
PRMSEQ    17E0     6583, 1115,10434
PRMTAB    7EE9    10548, 6584
PRNTAL    0010     67
PRO010    17F2     6607, 6598
PRO100    1814     6633, 6592
PROCSR    0070     470, 1127, 2890, 2892, 5646, 5648, 6389, 6393
PROFLD    FFC2     617, 623, 3677, 5090, 6183, 8533, 8541, 8649,10248

```

13255

2648A MICROCODE LISTING 'PT91'
SYMBOL VALUE REFERENCED ON

13255/90010
REV 04/17/7E

```
=====
```

PROMPT	000D	278, 5495
PRSTRT	17E6	6590, 6627, 10516
PRV100	0CBF	3996, 3976
PRV110	0CC2	3998, 4003
PRVPG1	0CBD	3983, 5396
PTB090	0741	2501, 2543
PTB100	0744	2503, 2465, 2567, 3765
PTB200	074E	2514, 2470, 2473
PTB220	076E	2538, 2520, 2533
PTB300	0789	2565, 2497
PTBLK	0710	2462, 1076, 2225, 6222
PTDLY	05DC	555
PTR120	0281	1433, 1418, 1425
PTRABT	FE78	942, 945
PTRBBG	FE7D	939, 940
PTRBD2	001F	568, 1428
PTRBLN	0100	584
PTRBPT	FE79	941, 942
PTRCF2	8540	518, 1427
PTRCL1	8D02	510, 1416
PTRDA2	8560	517
PTRDY1	0001	559
PTRDY2	0002	564
PTRFLG	FE77	945, 952, 1434
PTRHD2	00E0	567
PTRI10	026F	1422, 1415
PTR0L2	0020	566
PTROT1	8D20	508
PTROT2	8540	515, 1430
PTRP01	0080	560
PTRSB2	0040	565
PTRSPT	FE7B	940, 941
PTRST1	8D00	509, 1413
PTRST2	8520	516, 1423
PTTPLN	2832	989, 990, 2542
PUTBRK	0005	270, 5514
PUTLIN	078E	2584, 3431, 8014
QUOTE	0027	380
R	0052	404, 5462
RADIX	FFD4	182, 183, 1977, 5580
RAMERR	1065	4794, 4747
RC4010	07EC	2681, 2701
RCA120	0826	2766, 2771
RCA130	0832	2777, 2785
RCA140	083A	2782, 2760
RCA200	083F	2790
RCA210	084B	2801, 2806
RCA220	0859	2811, 2772, 2781
RCA240	0863	2821, 2753
RCA245	086E	2832, 2815
RCA250	0876	2840, 2825
RCA255	0880	2847, 2835
RCA260	0881	2854, 2845

13255

2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
RCA270 08A0 2869, 2867
RCA440 07F4 2689, 2672, 2678
RCA460 0807 2703, 2684
RCADDR 080D 2744, 5375, 9607
RCADR0 0810 2746, 2628, 3647, 4055, 5413, 6623, 9465
RCADR1 07B1 2620, 3416, 3498, 7010
RCADR2 07B5 2623, 2615
RCADR3 07B8 2625, 2657
RCADR4 07CA 2654, 5191, 7392, 7419, 7630, 7936, 10240
RCADRA 07A1 2608, 1099, 4227, 6890
RCADRB 07A9 2612, 8327
RCKYCD 009D 6092, 5681
RCRDGO 2826 982, 983, 2175
RCVMDE 0020 92, 1502
RDABRT 2837 991, 992, 1474
RDP010 1E83 8168, 8157, 8159, 8165
RDWOWT 0001 787
RDY 0040 777
REC 0008 841
RECINI 0010 792
RECKEY 280B 970, 971, 6117
RECORD 0040 125, 2169
RECPGE 0020 793
RECRWD 0008 790
RECSEP 5003 240, 241, 5784
REDKEY 2805 968, 969, 6116
RELSNS 0004 719, 5424, 5431, 5456
RELTAK FF61 826, 828
REMOTE 0008 133, 2122, 2413, 2414, 5872, 5902
REMSET 0010 91, 1502, 2125, 2198, 5098, 5680, 5881, 5915
RESET 0000 468, 3155, 6578
RET 00C9 449, 1218
REXMIT 0001 460
RGTCTU 0002 850
RHTMGN FFBE 626, 627, 628, 1630, 2377, 6757, 7200, 7236, 7359, 7397,
7475, 7663, 7849, 7865, 8183, 8794, 8950, 9197

RIGHT 0000 360, 8996
RIP 0004 781
RLCRSN 1302 5423, 10524
RLD080 0CF8 4046, 4018
RLD085 0D13 4064, 4049
RLD090 0D21 4078, 4039
RNGTA FFD2 183, 184, 1787, 1978, 2002, 4200
ROL080 0D6F 4142, 4093
ROL090 0D7D 4156, 4113
ROL100 0DSA 4116, 4159
ROL200 0D5C 4118, 4086
ROLLCT FF82 680, 690, 3838, 3844, 3997, 4000, 7003, 7038, 7043, 7076,
7094, 7104, 7110, 8440, 8446
ROLLDN 0CCE 4016, 3999, 4060, 7106, 7176, 10507
ROLLUP 0D30 4091, 2528, 3633, 3843, 5274, 8445, 10506
ROLUP1 0D77 4153, 8297
ROLUP2 0D5D 4120, 5281

```

13255

2648A MICROCODE LISTING 'PT91'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
ROLUP3  0D5F  4126, 8263
ROLUPC  0D60  4128, 8283
ROMERR  1061  4791, 4480
RPTKEY  0003  219
RSETDC  0002  267, 1154
RSETKB  0007  223, 1152
RSTCTU  2817  974, 978, 1159
RSTDSP  1E6A  8148, 1063, 1161, 4225
RSTJMP  0001  469, 1528, 9727
RSTOFF  0004  478, 1731, 1852
RSTON   0002  477, 4877
RSTTMR  FFD0  187, 1138, 2893, 4258
RTABLE  7DFF  10369, 1997, 4199
RTB010  7E1A  10390, 10373
RTB020  7E1C  10392
RTNKEY  162D  6136, 1342
RUN     0001  838, 6304, 6552
RXMERR  106C  4800, 4777
S       0053  405
SAVINP  FF23  919, 920
SAVOUT  FF22  920, 924
SBINRY  0002  718, 5125, 5911
SBL010  183D  6690, 6671
SBL020  183E  6692, 6686
SBLXF0  1825  6661, 1070, 1112, 4292, 5134, 5435, 10291
SBLXF1  1830  6682, 5701, 5727, 5975
SBLXFA  1828  6668, 1071, 5713, 5974
SCHRST  0D85  4167, 10438
SCHST1  0D8B  4173, 10584
SCNCNT  FF54  857, 858, 1483
SCNVEC  9168  145, 1220, 1487
SCRNRW  FFD9  684, 5308, 5327, 5357, 9060
SCRSEN  1000  710, 4353, 5434, 5438
SDACOM  0001  728, 1498, 1885, 1952, 4780, 5149, 5630, 6737, 6809, 8193
SDC2    0100  706, 5493, 5729, 6685
SDTER1  1351  5482, 5830
SDTERM  134E  5480, 1080, 1114, 4307, 6016, 6020, 10302
SDTRM1  1851  6713, 1081, 5481, 5828
SDTRM2  185A  6717
SDTRM3  1865  6723, 5794
SDVDUN  8000  713
SDVREC  0001  717, 5125, 5911
SDVST   0800  709
SELECT  0020  124
SELKEY  280E  971, 972, 6118
SENER   4000  712, 3638, 4353, 5683, 5700, 5834, 6647
SESCTB  7E29  10412, 1962
SETCH   0020  78
SETDF0  186A  6736, 2167, 6978, 9701, 10207
SETDFL  186C  6738, 5971, 6483, 6911
SEFRN   000C  228, 7952
SETLCL  0004  269, 5879
SETLFT  1872  6754, 10478
=====

```

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```
=====
SETMF2 1894 6785, 1957, 5339, 5425, 7340, 9899
SETMON 0008 273, 6028
SETNRM 0009 274, 6058
SETREM 0003 268, 5913
SETRHT 1884 6765,10479
SETROM 0000 115, 1003, 1216, 5645
SETTRG 0001 266, 1892
SETTRM 189A 6794, 6882,10518
SEVEN 0037 393, 1055
SFCTKY 2000 711, 4353, 5973, 6006
SFKCHK 10EC 4893, 8670
SFKYAT 00C8 440, 6130, 6138, 9845
SFKYDS 0DC6 4238, 1883,10365,10633
SFKYOF 0D96 4192, 4246, 4960, 6050, 6070, 8125,10535
SFKYON 0DAE 4212,10534
SFO010 0DBA 4223, 4198, 4201
SFTCNT FF5D 833, 834
SFTCR 00EF 481, 1469, 1545, 9742
SFTDLY 0032 462, 1141, 1145
SFTEND 0010 455,10226
SFTERR 0008 821
SFTKEY FE75 958, 9300, 9310
SFTKYS FFA6 638, 639, 1336, 5955, 8281, 9133
SFTRST 0DDE 4254,10531
SHFT1 0DF1 4268, 4285
SHFT2 0DF8 4274, 4670
SHFTIN 0DFD 4281,10407
SHFTOT 0DEA 4264,10406
SI 000F 373, 9824
SIX 0036 392, 1047
SKFLGS FE76 952, 958, 1151, 9286, 9319, 9337, 9364, 9373
SKIP 0002 955, 1559, 9302, 9305, 9343
SKPTRM 0008 731, 6868,10010
SLANT 002F 386
SLKYCD 009E 6093
SMALLA 0061 416, 5442, 9896
SMALLD 0064 418, 9780
SMALLF 0066 419, 9874
SMALLI 0069 420
SMALLK 006B 421, 9880
SMALLP 0070 422
SMALLX 0078 423, 1366
SNDATN 000B 276, 6820
SNDCD1 1885 6817,10659
SNDCD2 13FF 5665,10662
SNDCDE 18A7 6807,10570
SNDCTB 7F67 10654, 6811
SNDFCT 000C 277, 5668
SO 000E 372, 9820
SPLDIS 0002 36, 9265
SPOWL FF6C 743, 748, 1443, 1708, 3621, 7292, 8203, 9267, 9556
SPOWOF 00FF 746, 3622, 7293
SPOWON 0020 745, 9268
```

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'PT91'

SYMBOL VALUE REFERENCED ON

```
=====
```

SSTAT	0200	707, 4291, 4297
SSTAT2	0400	708, 4347, 10290, 10296
STA010	0E63	4365, 4363
STA2G1	7DC1	10313, 10301
STA2G2	7DC4	10315, 4731
STA2G0	7DAD	10295, 1935
STACK	9160	588, 1129, 1449
STAPAR	0E26	4318, 4302, 4720
START	0260	1404, 1163
STAT2	7DA7	10289, 10454
STATGO	0E0B	4296, 1934
STATUS	0E05	4290, 10517
STB050	18E9	6881
STB080	1904	6897, 6889
STB090	1915	6909, 6885
STBLMD	0004	220
STC010	0E8B	4407, 4401
STCHR1	0E79	4397, 2331, 3512
STCHST	000D	229, 5762
STCMFL	1544	5924, 3152, 4216, 5552, 6508, 8853
STFOR1	00FE	6109, 1585
STFOR2	00FD	6108, 1587
STOREA	1757	6475, 4082
STPFLG	00C4	436, 5228, 6797, 6883, 8064, 8360, 8506, 8633, 9491, 9496, 10045, 10266
STPR	00C0	432, 2306, 3672, 6128, 6138, 6591, 6626, 7465, 7687, 8062, 8337, 8337, 8379, 8529, 8529, 8648, 9544, 9570, 10066
STPRPT	0009	225, 9745
STPXFR	FFFF	739, 5759, 10253
STR010	12F0	5406, 5416
STRTAK	405F	830
STRTB1	18C4	6845, 6842
STRTBL	18BE	6841, 1072
STRTST	0005	221, 4436
STRXMO	17EB	6596, 10543
STSKFL	23FD	9363, 1560, 9117, 9306, 9350
STTERM	18D2	6865, 5692
SUPCHR	0020	343, 2145, 2162
SWAP	2317	9189, 1385, 4224, 9128, 9219
SWAP0	2311	9184, 9162, 9227
SWAP1	231D	9194, 2506
SWCHAR	000B	227, 4270
SWP010	2325	9201, 9211
SWPCTU	FF24	916, 919
SWPSTR	FFAF	628, 629, 9196
T	0054	406, 4901, 9159
TAK	0008	780
TCHAR	FF68	751, 752, 4664, 4672, 4678
TEMP	FF9D	650, 651, 7767, 7776, 8376, 8387, 8485
TEMP1	FF9E	649, 650
TEST	0E8E	4415, 10542
TESTOK	0002	97, 4717
THREE	0033	389, 1023

```
=====
```


13255
 2648A MICROCODE LISTING 'PT91'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
TKI      0080      776
TLINO    FFA3      640, 641, 4031, 4101, 4824, 5343, 5393, 5460, 8201,10070,
          10142
TMI010   08C7      2900, 2896
TMI020   08CF      2906, 2904
TMI100   08E0      2922, 2909
TMI110   08FC      2938, 2926
TMIACK   0000      107
TMIEN    0002      110, 1003, 1216, 2889, 2891
TMINTR   08A9      2884, 1024
TMIOFF   0020      112
TMPCOL   FF85      674, 678, 2747, 2813, 2822, 2865, 7028, 7091
TMRINT   0003      102, 2924, 5656
TMRON    0001      109, 1003, 1216
TOP100   10AF      4829, 4827
TOPLIN   FFCB      603, 606, 2375, 2515, 3729, 3739, 4023, 4065, 4103, 4117,
          4143, 4834, 7078, 8171, 8216, 8243
TOPUP1   10B0      4831, 3535
TOPUPD   10A3      4820, 4076, 4154
TPSTAL   FF50      862, 866, 2901
TRIGGR   5002      239, 240, 1887
TRMFACT  FF6D      737, 743, 2611, 2834, 2858, 5760, 5832, 6796, 7009, 8532,
          8538, 8681,10239,10254
TRMRDY   1094      4814, 1397
TRMTST   0E9D      4426, 1083
TRMTYP   FFFD      152, 153, 1233, 1262,10326
TST010   0EBC      4447, 4457, 4459, 4464, 4476
TST020   0EDF      4470, 4461
TST030   0EEA      4479, 4466
TST050   0EF9      4502, 4455
TST060   0F0A      4517, 4521
TST090   0F18      4531, 4601, 4614
TST100   0F18      4537, 4546
TST115   0F25      4549, 4552
TST120   0F2C      4557, 4567
TST125   0F39      4570, 4573
TST130   0F3E      4578, 4589
TST135   0F59      4602, 4598
TST140   0F61      4611, 4606
TST150   0F74      4635, 4643
TST160   0F87      4646, 4639
TST200   0F94      4660, 4688
TST220   0F9B      4665, 4682
TST240   0FA6      4671, 4684
TST420   0FD1      4695, 4706
TST440   0FE6      4708, 4703
TST500   1017      4742, 4644, 4649
TST510   1019      4745, 4561, 4583
TST600   101D      4759, 4489
TST610   1038      4773, 4765
TSTCTU   2811      972, 973, 4420
TWO      0032      388, 1015
TYPSET   10E0      4884,10442
  
```

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

=====

U	0055	407
UNIT0	FF63	805, 816
UNLKKB	0002	218, 6458
UP	0003	363, 9042
US	001F	349, 2157
USL	0010	842
USREAD	0002	788, 1473
VERIFY	0080	795
VERSN	0054	16, 1000, 2697, 4726, 8495
VERSN2	0055	7, 6619
VRTBAR	007C	425,10298
WBSR	0020	134
WRPDEL	0020	722, 7339, 7377
WRPFLG	0040	723, 1672, 1684, 8981
WRTERR	0020	823
WTL010	029A	1454, 1485, 1490, 1505
WTL020	0288	1479, 1461, 1471
WTL200	02D0	1495, 1463, 1648
WTL205	02E8	1507, 1503
WTL207	02F9	1518, 1515
WTL210	030D	1533, 1521
WTL220	0325	1552, 1546
WTL225	0331	1558, 1550
WTL230	033C	1563, 1554, 1556
WTL250	0345	1573, 1537, 1541
WTL260	0347	1575, 1594, 1599, 1604, 1609
WTL270	0368	1596, 1584
WTL280	036F	1601, 1586
WTL290	0375	1606, 1588
WTL300	037C	1615, 1511, 1567
WTL310	03A5	1637, 1617, 1621, 1624, 1628
WTL00P	029A	1448, 1517, 1529, 1551, 1562, 1591, 1641, 1644
XBF2DS	0080	734, 4432, 5630, 8980
XDS2BF	0020	800, 3641, 5067,10055,10166
XFRLIM	FF47	876, 877
XMD000	10FD	4917, 4934
XMD010	1100	4920, 4937, 4943
XMD020	111C	4941, 4932
XMD030	1122	4947, 4929
XMOHME	1955	6977, 5741, 6852,10495
XMONLY	00C2	434, 2690, 6597, 8566, 8637, 8934, 9478, 9831
XMS2DS	1102	4924, 1343, 1372, 8134
XPD001	192A	6934, 6958
XPD005	1935	6940, 6933
XPD010	1937	6948, 6939
XPD050	194B	6965, 6938
XPUTDC	1922	6929, 1082, 2425, 4301, 5443, 5447, 5451, 5474, 5764, 5771, 5785, 5793, 5807, 6021, 6716, 6719, 6725, 6926,10300
XTRASP	FE80	930, 932
Y	0059	408, 5458
Z	005A	409
ZALPCK	4823	207, 208, 8663
ZANCHK	600B	302, 303, 1729, 3290, 3390, 3630, 4107, 4863, 4872

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'PT91'

SYMBOL VALUE REFERENCED ON

```

=====
ZANCUR 6073 337, 338, 8997, 9008, 9026, 9043
ZAPCHK 6029 312, 313, 2408
ZAPCR 6056 327, 328,10404
ZAPFLG FB96 353, 2406, 8883
ZAPHME 606A 334, 335, 5739, 6848
ZAPLF 6076 338,10397
ZAPMOF 601A 307, 308, 4241
ZAPSCN 601D 308, 309, 8885
ZBELL 4814 202, 203, 1504, 1633, 3775, 4658, 5544, 6624, 6969, 8936,
9627, 9646, 9736, 9755,10391

ZBRK1 0800 2696, 4724, 2698
ZBRK1C 0802 2699, 2694
ZBRK2 1000 4725, 6614, 4727
ZBRK2C 1002 4728, 4723
ZBRK3 1800 6615, 8493, 6620
ZBRK3C 1802 6621, 6613
ZBRK4 2000 8494, 8496
ZBRK4C 2002 8497, 8492
ZBS 6047 322, 323, 8864
ZCHKTK 6070 336, 337
ZCLMD1 4811 201, 202, 3693, 6062, 6184, 8847
ZCLXMT 481A 204, 205
ZCR 6020 309, 310, 9261
ZCTLAL 9214 290, 291
ZDCBAS 5000 238, 239, 249
ZDCCTL 5011 252, 253, 5541, 6059
ZDCINT 5026 259, 5507
ZDCMON 500E 251, 252, 2933
ZDCTST 5014 253, 254, 5615
ZDPTST 604D 324, 325, 1705, 9710
ZDSPMS 0040 1062
ZERO 0030 387, 1359, 2972, 3046, 3066, 3087, 3963, 4888, 5577, 9850,
9878, 9887

ZGBASE 6000 298, 299
ZGCKEY 600E 303, 304
ZGETAL 920E 288, 289
ZGETDC 5017 254, 255, 2131, 6549
ZGETKY 4805 197, 198, 1462, 9739
ZGFLG1 90B2 351, 4841
ZGFLG6 9097 354, 1703, 3616, 8754, 8862, 9259, 9708
ZGFUNC 602C 313, 314, 1516
ZGGINT 6064 332, 333,10282
ZGRGET 6067 333, 334, 9926
ZGRTST 6061 331, 332, 9925,10200,10281
ZGSBLK 906B 352, 1920, 5129
ZGSOFT 6005 300, 301, 1157
ZGSTAT 605E 330, 331, 1922
ZGSTUP 6008 301, 302,10430
ZGTBIN 501D 256, 257
ZGTEST 606D 335, 336, 4653
ZHT 6041 320, 321, 8756
ZIN2AL 9205 285, 286, 1253
ZIN2DC 500B 250, 251, 1229

```

13255

2648A MICROCODE LISTING 'PT91'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
ZINFIX  6026    311,  312, 9595
ZINGR   6002    299,  300, 1391
ZINIAL  9202    284,  285, 1248
ZINIDC  5008    249,  250, 1227
ZINIKB  4802    196,  197, 1226
ZINTAL  9208    286,  287, 6398
ZKBBAS  4800    195,  196
ZKBCTL  4808    198,  199, 1153, 2899, 4271, 4419, 4437, 4738, 5677, 5763,
        6232, 6238, 6459, 6487, 6954, 7953, 9746
ZKBMON  4808    199,  200, 2931
ZLF     604A    323,  324, 3618
ZMONAL  9208    287,  288, 2930
ZMSGAL  921A    292
ZMUCHK  6023    310,  311, 1555, 1623, 1994, 2417, 2441, 4240, 4839, 5718,
        5738, 5797, 5818, 6847, 8158, 9861
ZMUTB   605C    329,  330, 1993
ZNDBIN  5023    258,  259
ZNUMCK  4826    208,  212, 8665
ZPAGE   6032    315,  316, 10459
ZPUTAL  9211    289,  290
ZPUTDC  501A    255,  256, 6937
ZRELGC  6011    304,  305
ZRETRN  0808    2709, 3678, 8328, 8640, 8668
ZSTAAL  9217    291,  292
ZSTBIN  5020    257,  258
ZSTGIN  6035    316,  317, 10461
ZSTJPR  481D    205,  206, 10559
ZSTLKY  4820    206,  207, 10553
ZSTMD1  480E    200,  201, 3710, 3773, 6040, 8185, 8841
ZSTXMT  4817    203,  204
ZTINTR  6014    305,  306
ZTKCLR  603E    319,  320, 1949, 10382
ZTKCUR  6038    318,  319, 10463
ZTKFLG  90AD    350, 2144, 2161, 2419
ZTKHC   6038    317,  318, 10465
ZTKSTR  6059    328,  329
ZTKSUP  602F    314,  315, 10380
ZVID1   6050    325,  326, 4218, 8098
ZVID2   6053    326,  327, 8162
ZVR     6017    306,  307, 1481
ZVT     6044    321,  322, 10398

```

1447 SYMBOLS, 4968 REFERENCES, -69 WORK TRACKS

```

=====
ITEM   LOC   OBJECT CODE  SOURCE STATEMENTS                                PAGE   1
=====
  1   0000   .   .   .   ASB,HEX ;GR70 21JUN77
  2   0000   .   .   .   ;*****
  3   0054   .   .   .   VERSN EQU 1240 ;GRAPHICS = VERSION 'T'
  4   0000   .   .   .   ;*****
  5   0000   .   .   .   ORG 60000Q ;START AT 24K
  6   6000   .   .   .   BEGIN EQU $
  7   6000   54   .   .   DB VERSN ;SET ROM PRESENT FLAGS
  8   6001   60   .   .   DB BEGIN/256
  9   6002   .   .   .   ;*****
10   6002   .   .   .   ; ENTRY VECTORS TO GRAPHICS
11   6002   .   .   .   ;*****
12   6002   C3   B9   62   JMP HRDRST ;HARD RESET
13   6005   C3   4F   63   JMP SFTRST ;SOFT RESET
14   6008   C3   7A   63   JMP GSETUP ;INITIATE GRAPHICS ESCAPE SE
15   600B   C3   B7   A2   JMP ANCHK ;A/N DISPLAY INHIBIT CHECK
16   600E   C3   35   A0   JMP GKKEYS ;CHECK FOR CURSOR KEY DOWN
17   6011   C3   84   A0   JMP RELGC ;CHK FOR CURSOR KEY RELEASED
18   6014   C3   B0   A1   JMP TINTR ;TIMER INTERRUPT
19   6017   C3   CB   A1   JMP VR ;VERTICAL RETRACE ROUTINES
20   601A   C3   D9   AA   JMP APMUOF ;TURN AUTP PLOT MENU OFF
21   601D   C3   6E   AF   JMP APSCAN ;AUTO PLOT FIELD SCAN
22   6020   C3   AF   9C   JMP XCR ;PROCESS CARRIAGE RET
23   6023   C3   EE   B8   JMP MUCK ;SEE IF AP MENU ON
24   6026   C3   33   B9   JMP INSFIX ;A.P. INSERT CHARACTER KLUGE
25   6029   C3   FA   B8   JMP APCHK ;AUTO PLOT CHECK FOR KYBD
26   602C   C3   71   A4   JMP KBFUNC ;PROCESS KEYBOARD FUNCS
27   602F   C3   62   69   JMP TKSTUP ;GS RECEIVED, SET UP FOR TEK
28   6032   C3   1D   6A   JMP PAGE ;TEK PAGE KEY FUNC (ESC FF)
29   6035   C3   DE   6A   JMP STGIN ;ESC-SUB RECEIVED, START GIN
30   6038   C3   2D   6C   JMP TKHC ;ESC-ETB MAKE TEK HARD COPY
31   603B   C3   6F   6A   JMP TEKAC ;RETURN TEK A/N CURSOR LOC
32   603E   C3   A9   6B   JMP CLRSUP ;CLEAR TEK ECHOPLEX SUPPRESS
33   6041   C3   9E   9D   JMP XHT ;PROCESS HT
34   6044   C3   AA   9D   JMP XVT
35   6047   C3   76   9D   JMP XBS
36   604A   C3   F9   9C   JMP XLF
37   604D   C3   69   63   JMP DSPTST ;TEST WHICH DISPLAY FOR CHAR
38   6050   C3   31   A4   JMP VIDEO1 ;GRAPHICS OFF,ALLOW A/N
39   6053   C3   4E   A4   JMP VIDEO2 ;RESTORE GRAPHICS, A/N
40   6056   C3   BE   AF   JMP APCR ;AUTO PLOT CARRIAGE RETURN
41   6059   C3   32   69   JMP TKSTRP ;CHECK TEK STRAPS
42   605C   F8   61   .   DW MUTB ;AUTO PLOT MENU RANGE TABLE
43   605E   C3   89   73   JMP STATGO ;SEND GRAPHICS STATUS
44   6061   C3   21   9E   JMP GGTEST ;TEST FOR GRAPHICS DATA GET
45   6064   C3   24   9E   JMP GGINIT ;INITIALIZE FOR GRAPHICS GET
46   6067   C3   37   9E   JMP GRGET ;GET GRAPHICS DATA
47   606A   C3   9B   AD   JMP HOME ;HOME AUTO PLOT CURSOR
48   606D   C3   E2   99   JMP GTEST ;GRAPHICS SELF TEST
49   6070   C3   56   69   JMP CHKTEK ;SEE IF IN TEK MODE
50   6073   C3   E5   9D   JMP ANCUR ;MOVE GC FROM A/N KEYS
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE  2
=====
52      6076      C3 D0 AF          JMP  APLF          ;AUTO PLOT LINE FEED
53      6079      . . .            ;*****
54      6079      . . .            ; KEYBOARD ROUTINES
55      6079      . . .            ;*****
56      4814      . . .            ZBELL EQU 44024Q   ;RING THE BELL
57      482C      . . .            ZLWASC EQU 44054Q  ;ADDRESS OF LOWER CASE TABLE
58      91EC      . . .            ZKBTMR EQU 110754Q ;REPEATING KEY TIMER)
59      4805      . . .            ZGETKY EQU 44005Q  ;GET KEYBOARD CHAR
60      4811      . . .            ZCLMD1 EQU 44021Q  ;CLEAR MODE 1 FLAG
61      4808      . . .            ZKBCTL EQU 44010Q  ;KEYBOARD CONTROL
62      FF12      . . .            ZCTCOL EQU 177422Q ;COLUMN KEY COLUMN
63      0001      . . .            ZCTLKY EQU 1Q      ;CONTROL KEY
64      FF0C      . . .            ZKBLED EQU 177414Q ;KEYBOARD LEDS
65      FF0E      . . .            ZBLFLG EQU 177416Q ;BLINK FLGS
66      0020      . . .            SELLED EQU 40Q     ;SELECT (GOLD) LED
67      6079      . . .            ;*****
68      6079      . . .            ; 2645 MAIN CODE ROUTINES FROM PT773
69      6079      . . .            ;*****
70      0040      . . .            ZMAIN EQU 100Q     ;BASE
71      0040      . . .            ZDSPMG EQU ZMAIN   ;DISPLAY MESSAGE
72      0043      . . .            ZRSTDP EQU ZMAIN+3Q ;RESTORE NORMAL DISPLAY
73      004F      . . .            ZESCND EQU ZMAIN+17Q ;TERMINATE ESCAPE SEQUENCE
74      0082      . . .            ZCHINT EQU ZMAIN+102Q ;CHARACTER INTERPRETATON
75      00B4      . . .            ZDELAY EQU 264Q    ;DELAY 10MS + L REG
76      00B7      . . .            ZESCAP EQU 267Q    ;SET UP FOR ESC SEQ
77      00BA      . . .            ZDSPCO EQU 272Q    ;ADD DISPLAY ENHANCEMENT
78      00BD      . . .            ZCRADV EQU 275Q    ;CLEAR CURSOR ADVANCE FLAG
79      00C0      . . .            ZCRRET EQU 300Q    ;DO CARRIAGE RETURN
80      0088      . . .            ZGETDP EQU 210Q    ;GET DISPLAY ROUTINE
81      008B      . . .            ZLNFD EQU 213Q     ;DO A LINE FEED
82      00C3      . . .            ZDCIO EQU 303Q     ;FLAG IF CHAR FROM DC OR IO
83      00C6      . . .            ZCHKSF EQU 306Q    ;SEE IF SOFT KEY MENU UP
84      007C      . . .            ZPUTDC EQU 174Q    ;SEND CHAR TO DATACOM
85      00C9      . . .            ZDSPCH EQU 311Q    ;ADD CHAR TO DISPLAY
86      00CC      . . .            ZB2DA EQU 314Q     ;CONVERT A REG TO DEC & XMIT
87      00CF      . . .            ZB2DDE EQU 317Q    ;CONVERT DE TO DEC & XMIT
88      00A8      . . .            ZBNDEC EQU 250Q    ;CONVERT DE TO BINARY
89      00AB      . . .            ZBNDC A EQU 253Q   ;CONVERT A TO BINARY
90      00D2      . . .            ZCKRMT EQU 322Q    ;SEE IF IN REMOTE
91      00D5      . . .            ZSBXFR EQU 325Q    ;SET BLOCK XFER FLAG
92      00D8      . . .            ZCLBXF EQU 330Q    ;CLEAR BLOCK XFER FLAG
93      00DB      . . .            ZSDTRM EQU 333Q    ;SEND TERMINATOR
94      00DE      . . .            ZPRMSQ EQU 336Q    ;ESC & RECEIVED
95      00E1      . . .            ZLCLN2 EQU 341Q    ;PROCESS CHAR
96      00E4      . . .            ZINITG EQU 344Q    ;INITIALIZE FOR DSPLAY GET
97      00E7      . . .            ZENTER EQU 347Q    ;ENTER--ESC D
98      00EA      . . .            ZHANG EQU 352Q     ;HANG THE TERMINAL
99      00ED      . . .            ZDPMG2 EQU 355Q    ;DISPLAY MESSAGE WITH G ON
100     00F0      . . .            ZCKCTL EQU 360Q    ;CHECK FOR CONTROL CODE
101     005B      . . .            ZSBXFA EQU 133Q    ;KEYBOARD INITIATED XFERS
=====

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC  OBJECT CODE  SOURCE STATEMENTS                PAGE   3
=====
  102    00A5  . . .      ZIORGO EQU 2450      ;EXECUTE ROUTINE IN IO ROM
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	4
104	6079	. . .	;*****		
105	6079	. . .	; 2645 MAIN CODE SYMBOLS		
106	6079	. . .	;*****		
107	FF88	. . .	ZCHAR EQU 177610Q		;LAST CHARACTER RECEIVED
108	FFD1	. . .	ZESCFG EQU 177721Q		;2-CHAR ESC SEQ FLAG
109	FFD2	. . .	ZRNGTA EQU 177722Q		;CURRENT RANGE TABLE
110	FFF1	. . .	ZMPT1 EQU 177761Q		;MESSAGE POINTER 1
111	FFEF	. . .	ZMPT2 EQU 177757Q		;MESSAGE POINTER 2
112	FFC0	. . .	ZCURROW EQU 177700Q		;CURRENT CURSOR ROW
113	FF69	. . .	ZLCHAR EQU 177551Q		;LAST CHARACTER PROCESSED
114	8720	. . .	ZIOCRW EQU 103440Q		;HW STROBE-CURSOR ROW
115	0017	. . .	ZMXROW EQU 23		;MAXIMUM VISIBLE CURSOR ROW
116	00CE	. . .	ZEOP EQU 316Q		;END OF PAGE CODE
117	FFC1	. . .	ZCURCL EQU 177701Q		;CURRENT CURSOR COLUMN
118	00CC	. . .	EOL EQU 314Q		;END OF LINE CODE
119	FBFF	. . .	ZDSPLM EQU 175777Q		;TOP OF DISPLAY MEM
120	0082	. . .	IVON EQU 202Q		;INVERSE VIDEO ON
121	0080	. . .	IVOFF EQU 200Q		;INVERSE VIDEO OFF
122	8700	. . .	ZIOCCCL EQU 103400Q		;STROBE FOR CURSOR COL
123	FFFE	. . .	ZDSPST EQU 177776Q		;DISPLAY START POINTER
124	FF89	. . .	ZDCHAR EQU 177611Q		;CHAR BEING DISPALYED
125	FFCB	. . .	ZTOPLN EQU 177713Q		;TOP LINE OF DISPLAY
126	FFA3	. . .	ZTLIN0 EQU 177643Q		;TOP ROW OF DISPLAY
127	FFD9	. . .	ZIOINP EQU 177731Q		;CURRENT FROM DEVICE
128	FFF4	. . .	ZMDFL1 EQU 177764Q		;SELECT MODE
129	0020	. . .	SELECT EQU 40Q		;CURRENTLY IN SELECT MODE
130	FFBF	. . .	ZLFTMG EQU 177677Q		;LEFT MARGIN
131	FF64	. . .	ZIOFL2 EQU 177544Q		;DISABLE ESC SEQ EXPANSION
132	0020	. . .	ZDS2BF EQU 40Q		;IN GETDSP
133	FFF1	. . .	ZMSGP1 EQU 177761Q		;MESSAGE POINTERS
134	FFEF	. . .	ZMSGP2 EQU 177757Q		
135	FFED	. . .	ZMSGP3 EQU 177755Q		
136	FFEB	. . .	ZMSGP4 EQU 177753Q		
137	0009	. . .	STPRPT EQU 11Q		;STOP KEY REPEAT
138	FFF3	. . .	ZMDFL2 EQU 177763Q		;MODE FLAGS 2
139	0008	. . .	REMOTE EQU 10Q		;IN REMOTE MODE
140	FE4F	. . .	DSPSTR EQU 177117Q		;MESSAGE BUFFER
141	FFFE	. . .	DISPST EQU 177776Q		;POINTER TO ST OF DISPLAY
142	FFFA	. . .	KBJMP2 EQU 177772Q		;KB JUMPERS #2
143	0020	. . .	PJMPR EQU 40Q		;STRAP P => SCALED TEK MODE
144	0040	. . .	QJMPR EQU 100Q		;STRAP Q => UNSCALED
145	FF9C	. . .	ZCHRIN EQU 177634Q		;ACTUAL INPUT CHARACTER
146	0001	. . .	DSPFNC EQU 1Q		;DISPLAY FUNCTIONS ON
147	FFFB	. . .	KBJMP1 EQU 177773Q		;KEYBOARD JUMPERS #1
148	0001	. . .	AJMPR EQU 1Q		;SEND CONTROL CODES STRAP
149	FF67	. . .	ZCAFLG EQU 177547Q		;CURSOR ADVANCE FLAG
150	FE77	. . .	PTRFLG EQU 177167Q		;TYPE OF PRINTER INSTALLED]
151	0005	. . .	PUTBRK EQU 5Q		;SEND BREAK
152	5011	. . .	ZDCCTL EQU 50021Q		;DATACOM CONTROL
153	FFF6	. . .	INTFLG EQU 177766Q		;INTERRUPT FLAG


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE   5
=====
154      0003      . . .      TMRINT EQU 30          ;TIMER INTERRUPT
155      0004      . . .      RSTOFF EQU 40         ;DISALLOW RESETS
156      0002      . . .      RSTON EQU 20          ;RE-ENABLE RESETS
157      8380      . . .      IOKBCO EQU 101600Q    ;KEYBOARD CONTROL FOR RESETS
158      282F      . . .      ZCTMON EQU 24057Q     ;MONITOR TAPES
159      8D20      . . .      PTROT1 EQU 106440Q    ;OUTPUT TO VIDEO PRINTER
160      6079      . . .      ;*****
161      6079      . . .      ; ALTERNATE I/O ENTRY VECTORS *
162      6079      . . .      ;*****
163      9200      . . .      ALTORG EQU 111000Q    ;ALTERNATE IO START (36.5K)
164      9202      . . .      ZINIAL EQU ALTORG+2   ;INITIALIZATION ROUTINE
165      9205      . . .      ZIN2AL EQU ZINIAL+3   ;INITIALIZATION CONTINUATOR
166      9208      . . .      ZINTAL EQU ZIN2AL+3   ;INTERRUPT PROCESSOR
167      920B      . . .      ZMONAL EQU ZINTAL+3   ;MONITORING ROUTINE
168      920E      . . .      ZGETAL EQU ZMONAL+3   ;INPUT ROUTINE
169      9211      . . .      ZPUTAL EQU ZGETAL+3   ;OUTPUT ROUTINE
170      9214      . . .      ZCTLAL EQU ZPUTAL+3   ;CONTROL ROUTINE
171      9217      . . .      ZSTAAL EQU ZCTLAL+3   ;STATUS ROUTINE
172      921A      . . .      ZMSGAL EQU ZSTAAL+3   ;ALTERNATE DEVICE NAME
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE   6
=====
174      6079      . . .      ;*****
175      6079      . . .      ; GRAPHICS EQUATES
176      6079      . . .      ;*****
177      6079      . . .      ; GFLGS1
178      0001      . . .      MOVE EQU 1Q          ;MOVE WITHOUT DRAWING VECTOR
179      0002      . . .      LINEPT EQU 2Q       ;USER-DEFINED LINE PAT ON
180      0004      . . .      AREAPT EQU 4Q       ;USER-DEFINED AREA PAT ON
181      0008      . . .      NEWWA EQU 10Q       ;USE NEW WRITE ADDRESS
182      0010      . . .      MSGON EQU 20Q       ;MESSAGE (FROM DSPMSG) ON
183      0020      . . .      AVINHB EQU 40Q      ;A/N VIDEO DISABLED
184      0040      . . .      ACINHR EQU 100Q     ;A/N CURSOR INHIBITED
185      0080      . . .      CLIPED EQU 200Q     ;VECTOR WAS CLIPPED
186      6079      . . .      ;*****
187      6079      . . .      ; GFLGS2 -- HARDWARE
188      6079      . . .      ;*****
189      0001      . . .      BUSY EQU 1Q         ;HW IS BUSY; INITIATE DRAW
190      0002      . . .      ZOOM EQU 2Q         ;TURN ZOOM MODE ON
191      0008      . . .      NEWZM EQU 10Q       ;CHANGE ZOOM PARAMETERS
192      0040      . . .      STBIT EQU 100Q     ;SELF TEST FAIL BIT
193      0010      . . .      DRWGC EQU 20Q       ;DRAW GRAPHICS CURSOR
194      6079      . . .      ;*****
195      6079      . . .      ; GFLGS3 -- GRAPHICS CURSOR STATUS
196      6079      . . .      ;*****
197      0001      . . .      SUPRO EQU 1Q        ;SUPRESS CURSOR
198      0002      . . .      SUPR1 EQU 2Q        ;SUPRESS CURSOR
199      0004      . . .      SUPR2 EQU 4Q        ;SUPRESS CURSOR
200      0008      . . .      TIMSUP EQU 10Q     ;TIMED CURSOR SUPRESS
201      0010      . . .      RBISON EQU 20Q     ;RB LINE ACTUALLY ON
202      0020      . . .      WANTRB EQU 40Q     ;USER WANTS RB FIGURE ON
203      0040      . . .      GCON EQU 100Q     ;G CURSOR ACTUALLY ON
204      0080      . . .      WANTGC EQU 200Q   ;USER WANTS G CURSOR ON
205      6079      . . .      ;*****
206      6079      . . .      ; GFLGS5 -- ZOOM AND MISCELLANEOUS
207      6079      . . .      ;*****
208      0001      . . .      GCM1 EQU 1Q        ;G-CURSOR MOVED FOR ZOOM
209      0002      . . .      WANTZM EQU 2Q      ;USER WANTS ZOOM ON
210      0004      . . .      SUPRZM EQU 4Q      ;SUPRESS ZOOM
211      0008      . . .      DWFRST EQU 10Q    ;DRAW-FIRST-DOT FLAG
212      0010      . . .      RBDRW EQU 20Q     ;CURRENTLY DRAWING RB LINE
213      0020      . . .      GCM3 EQU 40Q      ;G-CURSOR MOVED FOR RB
214      0040      . . .      GCM4 EQU 100Q    ;G-CURSOR MOVED FOR TEXT
215      0080      . . .      NWZOOM EQU 200Q   ;ZOOM PARAMETERS HAVE CHANGE
216      6079      . . .      ;*****
217      6079      . . .      ; TKFLGS--FLAGS FOR TEKTRONIX MODE
218      6079      . . .      ;*****
219      0001      . . .      UNSCLD EQU 1Q     ;UNSCALED TEK MODE ON
220      0002      . . .      MARG1 EQU 2Q      ;AT MARGIN 1
221      0004      . . .      XNEXT EQU 4Q      ;HI X BYTE IS NEXT
222      0008      . . .      GSMODE EQU 10Q    ;GS RECEIVED
223      0010      . . .      GINMOD EQU 20Q    ;IN TEK GIN MODE
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	7
224	0020	.	SUPCHR EQU 400 ;ECHOPLEX SUPRESS IN GIN MOD		
225	0040	.	SCLD EQU 100Q ;SCALED TEK MODE ON		
226	6079	.	*****		
227	6079	.	; GFLGS6--TEXT FLAGS		
228	6079	.	*****		
229	0001	.	SLANT EQU 1Q ;SLANTED CHARACTERS ON		
230	0002	.	GTEXT EQU 2Q ;GRAPHICS TEXT MODE		
231	0004	.	NOSUL EQU 4Q ;DONT UPDATE START OF LINE		
232	0008	.	CNTR EQU 10Q ;CENTER TEXT LINE		
233	0010	.	RTJUST EQU 20Q ;RIGHT JUSTIFY TEXT LINE		
234	0020	.	MIDCH EQU 40Q ;MIDDLE OF TEXT		
235	0040	.	TOPCH EQU 100Q ;TOP OF TEXT		
236	0080	.	LABEL EQU 200Q ;LABEL (ESC*L)		
237	6079	.	*****		
238	6079	.	; GFLGS7--PARAMETERS. USES AUTOPLLOT EQUATES		
239	6079	.	; NIP,HAVED,HAVEP		
240	6079	.	*****		
241	6079	.	;NIP EQU 1B NUMBER IN PROGRESS		
242	6079	.	;HAVEP EQU 10B HAVE DEC POINT		
243	6079	.	;HAVED EQU 20B HAVE DIGIT		
244	0002	.	APLABL EQU 2Q ;AUTOPLLOT LABEL IN PROGRESS		
245	0020	.	MINUS EQU 40Q ;HAVE MINUS SIGN		
246	0040	.	ASCII EQU 100Q ;ASCII VECTOR IN PROGRESS		
247	0080	.	RESET EQU 200Q ;TERMINAL HAS BEEN HARD RESE		
248	6079	.	*****		
249	6079	.	; GRAPHICS CONTROLLER EQUATES		
250	6079	.	*****		
251	8920	.	HWSTAT EQU 104440Q ;HARDWARE STATUS		
252	8960	.	GRESET EQU 104540Q ;HW RESET FOR CONTROLLER		
253	8961	.	VRESET EQU 104541Q ;RESET VERT RETRACE FLAG		
254	6079	.	;VECTOR PARAMETERS		
255	891E	.	D1 EQU 104436Q ;R0		
256	891C	.	D2 EQU 104434Q ;R1		
257	891A	.	M1 EQU 104432Q ;R2		
258	8918	.	SIGNM1 EQU 104430Q ;R3		
259	8916	.	M2 EQU 104426Q ;R4		
260	8914	.	SIGNM2 EQU 104424Q ;R5		
261	8912	.	DC EQU 104422Q ;R6--DOT COUNT		
262	8910	.	INITD EQU 104420Q ;R7--INITIAL D		
263	8911	.	MSBD EQU 104421Q ;MSBYTE OF D		
264	890E	.	LSBWA EQU 104416Q ;R8--WA BITS 0-11		
265	890C	.	MSBWA EQU 104414Q ;R9--WA BITS 12-17		
266	8902	.	VDC EQU 104402Q ;R14--VECTOR DRAWING DOT CNT		
267	6079	.	;SOFTWARE FLAGS		
268	8908	.	SELWA EQU 104413Q ;R10--SELECT OLD/NEW WA		
269	8909	.	SLFTST EQU 104411Q ;R11--SELF TEST		
270	8907	.	CONTST EQU 104407Q ;R12--CONTINUE SELF TEST		
271	8901	.	DRWDOT EQU 104401Q ;R15--DRAW FIRST DOT		
272	6079	.	;ZOOM		
273	8904	.	DCNTRL EQU 104404Q ;R13LO--DISPLAY CONTROL		

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  8
=====
274  8905  . . .  PRESHF EQU  104405Q  ;R13HI--PRE SHIFT
275  8906  . . .  ZOOMWC EQU  104406Q  ;R12--ZOOM WORD COUNT
276  8908  . . .  ZOOMRC EQU  104410Q  ;R11--ZOOM REPEAT COUNT
277  8912  . . .  ZALO EQU  104422Q  ;R6--ZOOM START ADDRESS 0-11
278  8910  . . .  ZAHI EQU  104420Q  ;R7--ZOOM START ADDRESS 12-1
279  6079  . . .  ;GRAPHICS CURSOR
280  891E  . . .  GC1DC EQU  104436Q  ;R0--DOT COUNT,HORIZONTAL
281  891A  . . .  GC1LO EQU  104432Q  ;R2--WALO, HORIZONTAL
282  8918  . . .  GC1HI EQU  104430Q  ;R3--WAHI, HORIZONTAL
283  891C  . . .  GC2DC EQU  104434Q  ;R1--DOT COUNT, VERTICAL
284  8916  . . .  GC2LO EQU  104426Q  ;R4--WALO, VERTICAL
285  8914  . . .  GC2HI EQU  104424Q  ;R5--WAHI, VERTICAL
286  6079  . . .  ;REGISTERS
287  8920  . . .  HWFLGS EQU  104440Q  ;LOAD F1,F3,F4,F5
288  8941  . . .  HCEJK EQU  104501Q  ;LOAD HCEJK AND SAMPLE
289  8940  . . .  PATERN EQU  104500Q  ;LOAD PATTERN BYTE
290  8921  . . .  SCALER EQU  104441Q  ;LOAD PRESCALER
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
292	6079	. . .	;*****	9
293	6079	. . .	; MORE GRAPHICS EQUATES	
294	6079	. . .	;*****	
295	0009	. . .	CLRMEM EQU 110 ;HCEJK BITS TO CLEAR MEMORY	
296	000A	. . .	SETMEM EQU 120 ;HCEJK BITS TO SET MEMORY	
297	0004	. . .	PATENB EQU 40 ;BIT IN HCEJK TO ENABLE PAT	
298	0010	. . .	GVENAB EQU 200 ;BIT IN HCEJK TO ENABLE VIDE	
299	8000	. . .	B15 EQU 1000000 ;MSBIT	
300	0020	. . .	VRFLAG EQU 400 ;VERTICAL RETRACE FLAG	
301	0009	. . .	MAXTYP EQU 9 ;MAX LINE TYPE +1	
302	0008	. . .	PNTPLT EQU 8 ;POINT PLOT LINE TYPE	
303	000D	. . .	CR EQU 150 ;CARRIAGE RETURN	
304	0010	. . .	MAXMAG EQU 16 ;MAX ZOOM MAGNIFICATION	
305	0087	. . .	LWRFUN EQU 2070 ;BASE OF KEYPAD FUNCTIONS	
306	0002	. . .	SNDNIL EQU 20 ;TEK GIN TERMINATOR	
307	0001	. . .	SNDEOT EQU 10 ;TEK GIN TERMINATOR	
308	00EF	. . .	SFTCR EQU 3570 ;SOFT RETURN KEY CODE	
309	0084	. . .	MAXLBL EQU 132 ;MAX LABEL (ESC * L) LEN	
310	0000	. . .	XMARG0 EQU 0 ;TEK MARGIN 0 X COORD	
311	0103	. . .	XMARG1 EQU 259 ;TEK MARGIN 1 COORD	
312	015E	. . .	YTEKHM EQU 350 ;TEK HOME Y COORD	
313	0008	. . .	MAXPRM EQU 8 ;MAX NUMBER OF PARAMETERS	
314	FFFD	. . .	SLOVEC EQU -3 ;DRAW 3 DOTS/SCAN LINE-ZOOM	
315	FFFC	. . .	NRMVEC EQU -4 ;DRAW 4 DOTS/LINE NORMALLY	
316	FF06	. . .	FSTVEC EQU -250 ;DRAW 250 DOTS WHEN VIDEO OF	
317	0096	. . .	ANGKEY EQU 2260 ;TEXT ANGLE KEY	
318	0097	. . .	SIZKEY EQU 2270 ;TEXT SIZE KEY	
319	0020	. . .	LWRCSE EQU 400 ;LOWER CASE CHAR	
320	0092	. . .	MUKEY EQU 2220 ;TOGGLE MENU KEY	
321	008B	. . .	ZINKEY EQU 2130 ;ZOOM IN KEY	
322	008C	. . .	ZOUTKY EQU 2140 ;ZOOM OUT KEY	
323	0088	. . .	CURKEY EQU 2100 ;TOGGLE CURSOR KEY	
324	0030	. . .	ZERO EQU 600 ;ASCII 0	
325	0031	. . .	ONE EQU 610 ;ASCII 1	
326	002C	. . .	CMMA EQU 540	
327	002E	. . .	POINT EQU 560	
328	002B	. . .	PLUS EQU 530	
329	002D	. . .	NEG EQU 550	
330	000C	. . .	MXSTAT EQU 12 ;MAXIMUM STATUS PARAMETER	
331	0001	. . .	IDBLOK EQU 1 ;VALUE TO SEND TERMINAL ID	
332	0004	. . .	GCWBLK EQU 40 ;READ GC POSITION WITH WAIT	
333	0007	. . .	XCELL EQU 7 ;X CHAR CELL SIZE	
334	000A	. . .	YCELL EQU 10 ;Y CHAR CELL SIZE	
335	0013	. . .	CLRSMP EQU 230 ;SAMPLE OFF, VIDEO ON	
336	0033	. . .	SETSMP EQU 630 ;SAMPLE ON, VIDEO ON	
337	0001	. . .	EOD EQU 10 ;END OF DISPLAY	
338	001B	. . .	ESC EQU 330 ;CODE FOR ESCAPE	
339	0001	. . .	UDLINE EQU 1 ;USER DEFINED LINE PAT	
340	0002	. . .	UDAREA EQU 2 ;USER DEFINED AREA PAT	
341	0001	. . .	LTXMIN EQU 10 ;X COORD IS LESS THAN MIN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 10
=====
342      0002      . . .      GTXMAX EQU 20      ;X COORD IS GREATER THAN MAX
343      0004      . . .      LTYMIN EQU 40      ;Y COORD IS LESS THAN MIN
344      0008      . . .      GTYMAX EQU 100     ;Y COORD IS GREATER THAN MAX
345      0001      . . .      PFBRAK EQU 10      ;DO TEK PAGE FULL BREAK
346      0002      . . .      PFBUSY EQU 20      ;DO TEK PAGE FULL BUSY
347      008D      . . .      CLRKEY EQU 2150    ;CLEAR KEY CODE
348      0081      . . .      ACBLOK EQU 2010    ;ESC-ENG IN A/N
349      0082      . . .      CPBLOK EQU 2020    ;ESC-ENG IN GRAPHICS
350      0083      . . .      GINBLK EQU 2030    ;ESC-SUB
351      0084      . . .      GCBLOK EQU 2040    ;ESC-SUB ESC-ENG
352      00A1      . . .      GCKEY EQU 2410     ;LOWEST CURSOR KEY CODE
353      00A5      . . .      FSTKEY EQU 2450     ;CODE FOR FAST KEY
354      0040      . . .      HISPD EQU 100Q     ;SPEED IN FAST MODE
355      0018      . . .      LOSPD EQU 30Q      ;SPEED IN NORMAL MODE
356      0010      . . .      INTDLY EQU 20Q     ;INITIAL REPEAT DELAY
357      0014      . . .      TIMEOUT EQU 20     ;GC SUPPRESS TIME OUT=200MS
358      0014      . . .      LEN EQU 20         ;CURSOR LENGTH
359      FFEC      . . .      MLEN EQU -20
360      0001      . . .      PINIT EQU 10      ;PRINTER INITIALIZATION
=====
  
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 11
362	6079	. . .	;*****	
363	6079	. . .	; GRAPHICS STORAGE--ALL IN FAST RAM ON	
364	6079	. . .	; SECOND ROM BOARD	
365	6079	. . .	;*****	
366	9000	. . .	FSTRM2 EQU 110000 ;BASE OF 2ND RAM = 36K	
367	6079	. . .	; FLOATING POINT ROUTINES NEED 64 BYTES	
368	6079	. . .	; OTHER GRAPHICS USE 128 BYTES (256 TOTAL AVAIL)	
369	9000	. . .	SCR EQU FSTRM2 ;FLOATING POINT STORAGE	
370	6079	. . .	; FLOATING POINT RAM 110000B-110100B	
371	0090	. . .	SCRB EQU SCR/256 ;BANK FOR FLOAT. POINT	
372	90E0	. . .	GRAM EQU FSTRM2+224 ;OTHER GRAPHICS	
373	6079	. . .	;	
374	6079	. . .	;	
375	90DE	. . .	XCURR EQU GRAM-2 ;CURRENT POINT	
376	90DC	. . .	YCURR EQU XCURR-2	
377	90DA	. . .	XNEW EQU YCURR-2 ;NEW POINT	
378	90D8	. . .	YNEW EQU XNEW-2	
379	90D6	. . .	DELTA EQU YNEW-2 ;NEWPOINT-CURRENTPOINT	
380	90D4	. . .	DELTAY EQU DELTA-2	
381	90D3	. . .	NEWCD EQU DELTAY-1 ;BOUNDS CODE, NEW POINT	
382	90D2	. . .	CURCD EQU NEWCD-1 ;BOUNDS CODE,CURRENT POINT	
383	90D1	. . .	OCTANT EQU CURCD-1 ;OCTANT OF VECTOR	
384	90CF	. . .	NEWGCX EQU OCTANT-2 ;NEW COORDS OF G CURSOR	
385	90CD	. . .	NEWGCY EQU NEWGCX-2	
386	90CB	. . .	CURGCX EQU NEWGCY-2 ;CURRENT COORDS OF G CURSOR	
387	90C9	. . .	CURGCY EQU CURGCX-2	
388	90B9	. . .	PRMBUF EQU CURGCY-16 ;BUFFER FOR PARAMETERS	
389	90B7	. . .	TMPBUF EQU PRMBUF-2 ;TEMPORARY DATA BUFFER	
390	90B6	. . .	PRMDEX EQU TMPBUF-1 ;PARAMETER BUFFER INDEX	
391	90B5	. . .	CURMOD EQU PRMDEX-1 ;CURRENT DRAWING MODE(HCEJK)	
392	90B4	. . .	CURPAT EQU CURMOD-1 ;CURRENT PATTERN BYTE	
393	90B3	. . .	SCALE EQU CURPAT-1 ;CURRENT PRESCALER VALUE	
394	90B2	. . .	GFLGS1 EQU SCALE-1	
395	90B1	. . .	GFLGS2 EQU GFLGS1-1 ;HW FLAGS	
396	90B0	. . .	GFLGS3 EQU GFLGS2-1 ;GRAPHICS CURSOR STATUS	
397	90AF	. . .	GFLGS4 EQU GFLGS3-1 ;ACTIVE CURSOR KEYS	
398	90AE	. . .	GFLGS5 EQU GFLGS4-1 ;ZOOM AND MISCELLANEOUS	
399	90AD	. . .	TKFLGS EQU GFLGS5-1 ;TEK MODE FLAGS	
400	90AC	. . .	XLOW EQU TKFLGS-1 ;TEK PARAMETERS FOR VECTOR	
401	90AB	. . .	XHI EQU XLOW-1 ;ENDPOINT	
402	90AA	. . .	YLOW EQU XHI-1	
403	90A9	. . .	YHI EQU YLOW-1	
404	90A8	. . .	LPAT EQU YHI-1 ;USER LINE PATTERN	
405	90A7	. . .	LSCALE EQU LPAT-1 ;LINE PATTERN SCALE FACTOR	
406	90A6	. . .	STFLAG EQU LSCALE-1 ;SELF TEST FLAG	
407	6079	. . .	;*****	
408	6079	. . .	; AUTO PLOT STORAGE	
409	90A2	. . .	XSCALE EQU STFLAG-4 ;X SCALE FACTOR	
410	909E	. . .	YSCALE EQU XSCALE-4 ;Y SCALE FACTOR	
411	6079	. . .	;*****	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	12
412	909C	. . .	PRMVEC EQU YSCALE-2 ;JMP ADDR FOR PLOT PARAMETER		
413	909A	. . .	XORG EQU PRMVEC-2 ;RELOCATABLE ORIGIN X COORD		
414	9098	. . .	YORG EQU XORG-2 ;RELOC ORIGIN Y COORD		
415	9097	. . .	GFLGS6 EQU YORG-1 ;GRAPHICS TXT FLAGS		
416	9096	. . .	GFLGS7 EQU GFLGS6-1 ;PARAMETER FLAGS		
417	9093	. . .	CFM1 EQU GFLGS7-3 ;M1 FOR DRAWING CHAR		
418	9091	. . .	CFXINC EQU CFM1-2 ;CHARACTER FILL INCREMENT		
419	908F	. . .	CFYINC EQU CFXINC-2		
420	908D	. . .	XCHSIZ EQU CFYINC-2 ;CHARACTER DIMENSIONS		
421	908B	. . .	YCHSIZ EQU XCHSIZ-2		
422	9089	. . .	XCHINC EQU YCHSIZ-2 ;CHARACTER SPACING INCREMENT		
423	9087	. . .	YCHINC EQU XCHINC-2		
424	9085	. . .	XLFINC EQU YCHINC-2 ;LINE FEED INCREMENT		
425	9083	. . .	YLFINC EQU XLFINC-2		
426	9081	. . .	CHPAT EQU YLFINC-2 ;POINTER TO CURRENT CHAR PAT		
427	907F	. . .	RBX EQU CHPAT-2 ;RBLINE X COORD		
428	907D	. . .	RBX EQU RBX-2 ;RB LINE Y COORD		
429	907B	. . .	XSOL EQU RBX-2 ;START OF G TEXT LINE		
430	9079	. . .	YSOL EQU XSOL-2		
431	9077	. . .	XCHADJ EQU YSOL-2 ;ADJUSTMENT FOR TOP OR MIDDLE		
432	9075	. . .	YCHADJ EQU XCHADJ-2 ;OF CHAR		
433	9074	. . .	LBLCTR EQU YCHADJ-1 ;LABEL LENGTH COUNT		
434	9072	. . .	XMIN EQU LBLCTR-2 ;MIN X CLIP LIMIT		
435	9070	. . .	XMAX EQU XMIN-2 ;MAX X CLIP LIMIT		
436	906E	. . .	YMIN EQU XMAX-2 ;MIN Y CLIP LIMIT		
437	906C	. . .	YMAX EQU YMIN-2 ;MAX Y CLIP LIMIT		
438	906B	. . .	GSBLOK EQU YMAX-1 ;GRAPHICS STATUS BLOCK		

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
440	6079	.	.	;*****
441	6079	.	.	; TEMPORARY STORAGE--THIS IS A COMMON SCRATCH
442	6079	.	.	; AREA USED BY SEVERAL ROUTINES. MUST INSURE THAT
443	6079	.	.	; ONLY ONE AT A TIME USES VARIABLES IN THIS AREA
444	6079	.	.	;*****
445	906B	.	.	TEMP EQU GSBLOK
446	6079	.	.	; AREA FILL AND GRAPHICS CURSOR
447	9069	.	.	XLEFT EQU TEMP-2 ;LEFTMOST X COORD, AREA & GC
448	9067	.	.	YBOT EQU XLEFT-2 ;BOTTOM MOST Y, AREA & GC
449	9065	.	.	YBOT45 EQU YBOT-2 ;YBOT * 45
450	9063	.	.	HEIGHT EQU YBOT45-2 ;HEIGHT OF AREA FILL
451	9061	.	.	GCX EQU HEIGHT-2 ;CURSOR STORE
452	905F	.	.	GCY EQU GCX-2
453	905D	.	.	ZXTEMP EQU GCY-2 ;ZX-360/MAG
454	6079	.	.	; CLIPPING AND VECTORS
455	9069	.	.	XSTART EQU TEMP-2 ;STARTING COORD OF VECTOR
456	9067	.	.	YSTART EQU XSTART-2
457	9065	.	.	XFIN EQU YSTART-2 ;ENDING POINT OF VECTOR
458	9063	.	.	YFIN EQU XFIN-2
459	9061	.	.	XMID EQU YFIN-2 ;CLIPPING MIDPOINT
460	905F	.	.	YMID EQU XMID-2
461	905D	.	.	XDEL EQU YMID-2 ;CLIPPING DELTA X,Y
462	905B	.	.	YDEL EQU XDEL-2
463	9059	.	.	XTEMP EQU YDEL-2 ;TEMPORARY STORE
464	9057	.	.	YTEMP EQU XTEMP-2
465	9055	.	.	TEMPDX EQU YTEMP-2 ;TEMPORARY STORE FOR DELTA
466	9053	.	.	TEMPDY EQU TEMPDX-2
467	6079	.	.	; SELF TEST
468	9051	.	.	VECCNT EQU TEMPDY-2 ;VECTOR COUNT
469	6079	.	.	; CHARACTERS
470	906A	.	.	CNT1 EQU TEMP-1 ;LOOP COUNTERS
471	9069	.	.	CNT2 EQU CNT1-1

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 14
=====
473      6079      . . .      ;*****
474      6079      . . .      ; GRAPHICS SLOW RAM
475      6079      . . .      ; IN DISPLAY MEMORY AT END OF MAIN CODE STORAGE
476      6079      . . .      ; MUST CHANGE MAIN CODE EQUATE DSPLIM TO REFLECT
477      6079      . . .      ; NEW DISPLAY MEMORY LIMITS
478      6079      . . .      ;*****
479      FBF7      . . .      HAPAT EQU ZDSPLM-8 ;USER 8X8 AREA PAT, HORIZ
480      FBEF      . . .      VAPAT EQU HAPAT-8  ;USER 8X8 AREA PAT, VERT
481      FBEE      . . .      SPEED EQU VAPAT-1  ;CURRENT CURSOR SPEED
482      FBED      . . .      GCTIMR EQU SPEED-1 ;CURSOR INITIAL DELAY TIMER
483      FBEC      . . .      SUPTMR EQU GCTIMR-1 ;CURSOR SUPRESS TIMER
484      FBEA      . . .      P360M EQU SUPTMR-2 ;+360/ZOOM MAGNIFICATION
485      FBE8      . . .      P180M EQU P360M-2  ;+180/MAG
486      FBE6      . . .      M360M EQU P180M-2  ;-360/MAG
487      FBE4      . . .      M180M EQU M360M-2  ;-180/MAG
488      FBE3      . . .      MAXSPD EQU M180M-1 ;MAX CURSOR SPEED
489      FBE2      . . .      DCBYTE EQU MAXSPD-1 ;DISPLAY CONTROL BYTE
490      FBE1      . . .      MAG EQU DCBYTE-1  ;ZOOM MAGNIFICATION
491      FBDF      . . .      ZX EQU MAG-2       ;X ZOOM COORD
492      FBDD      . . .      ZY EQU ZX-2        ;Y ZOOM COORD
493      FBDC      . . .      TEKTRM EQU ZY-1    ;TEK GIN TERMINATOR
494      FBDB      . . .      TANG EQU TEKTRM-1  ;GRAPHICS TEXT ANGLE
495      FBDA      . . .      TXMAG EQU TANG-1   ;GRAPHICS TEXT SIZE
496      FBD8      . . .      CHLEN EQU TXMAG-2  ;CHARACTER VECTOR LENGTH
497      FBD6      . . .      XGINSV EQU CHLEN-2 ;SAVE TEK GIN CURSOR
498      FBD4      . . .      YGINSV EQU XGINSV-2 ;LOCATION
499      FBD3      . . .      TXORG EQU YGINSV-1 ;LORG VALUE
500      FBD1      . . .      ILEN EQU TXORG-2   ;IMAGE MODE LINE LENGTH
501      FBCF      . . .      IMGX EQU ILEN-2    ;X IMAGE COORD
502      FBCE      . . .      IMGY EQU IMGX-2    ;Y IMAGE COORD
503      FBCB      . . .      IOFSTX EQU IMGY-2  ;X IMAGE OFFSET
504      FBC9      . . .      IOFSTY EQU IOFSTX-2 ;Y IMAGE OFFSET
505      FBC8      . . .      GGFLGS EQU IOFSTY-1 ;GRAPHICS GET FLAGS
506      FBC7      . . .      PAT2 EQU GGFLGS-1  ;PATTERN FOR AUTOPLT
507      FBC5      . . .      GETPTR EQU PAT2-2  ;BUFFER POINTER FOR GRA GET
508      FBC4      . . .      TEKPF EQU GETPTR-1 ;TEK PAGE FULL STRAPS
509      FBC4      . . .      GLAST EQU TEKPF
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 15
511	6079	.	*****	
512	6079	.	; AUTO PLOT VARIABLE STORE--ALL IN DISPLAY MEM.	
513	6079	.	; ADD AT TOP OF DISPLY MEM, AND CHANGE THE	
514	6079	.	; SYMBOL DSPLIM IN 2645 CODE TO REFLECT NEW	
515	6079	.	; DISPLAY MEMORY LIMIT	
516	6079	.	*****	
517	6079	.	; BUFFERS FOR AUTO PLOT MENU PARAMETERS	
518	FBC3	.	APB1 EQU GLAST-1 ;INTEGER	
519	FBC2	.	APB2 EQU APB1-1 ;INT	
520	FBC1	.	APB3 EQU APB2-1 ;INT	
521	FBC0	.	APB4 EQU APB3-1 ;INT	
522	FBBC	.	APB5 EQU APB4-4 ;FLOATING POINT	
523	FBB8	.	APB6 EQU APB5-4 ;FP	
524	FBB4	.	APB7 EQU APB6-4 ;FP	
525	FBBO	.	APB8 EQU APB7-4 ;FP	
526	FBAC	.	APB9 EQU APB8-4 ;FP	
527	FBA8	.	APB10 EQU APB9-4 ;FP	
528	FBA4	.	APB11 EQU APB10-4 ;FP	
529	FBA0	.	APB12 EQU APB11-4 ;FP	
530	FB9E	.	APB13 EQU APB12-2 ;INT	
531	FB9C	.	APB14 EQU APB13-2 ;INT	
532	FB9A	.	APB15 EQU APB14-2 ;INT	
533	FB98	.	APB16 EQU APB15-2 ;INT	
534	FB97	.	APFLG2 EQU APB16-1 ;MORE AUTO PLOT FLAGS	
535	FB96	.	APFLGS EQU APFLG2-1 ;AUTO PLOT FLAGS	
536	6079	.	*****	
537	6079	.	; AUTO PLOT SCAN VARIABLES	
538	6079	.	*****	
539	FB94	.	BGNCUR EQU APFLGS-2 ;CURSOR LOC AT START OF NUM	
540	FB93	.	NUMLEN EQU BGNCUR-1 ;LENGTH OF NUMBER BEING BUIL	
541	FB91	.	NUMPTR EQU NUMLEN-2 ;POINTER TO NUMBER BUFFER	
542	FB0D	.	NUMBUF EQU NUMPTR-MAXLBL ;BUFFER FOR ASCII DATA	
543	6079	.	*****	
544	6079	.	; NUMBUF WILL ALSO BE USED AS LABEL BUFFER	
545	FB0D	.	LBLBUF EQU NUMBUF	
546	6079	.	*****	
547	FB0C	.	COLCNT EQU NUMBUF-1 ;CURRENT DATA COLUMN	
548	FB0A	.	SKPCNT EQU COLCNT-2 ;NO. OF LINES TO SKIP	
549	FB08	.	PNTCNT EQU SKPCNT-2 ;NO. OF POINTS TO PLOT	
550	FB06	.	CURSAV EQU PNTCNT-2 ;SAVE CURSOR LOCATION	
551	FB05	.	IGNCNT EQU CURSAV-1 ;NO. OF CHARS TO IGNORE	
552	FB04	.	INSERT EQU IGNCNT-1 ;INSERTED CHARACTER COUNT	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 16
=====
554      6079      . . .      ;*****
555      6079      . . .      ; MENU VARIABLES
556      6079      . . .      ;*****
557      FB03      . . .      COL1 EQU INSERT-1 ;FIRST COL OF CURRENT FIELD
558      FB02      . . .      MUF LD EQU COL1-1 ;CURRENT FIELD
559      FB00      . . .      TOPSAV EQU MUF LD-2 ;SAVE TOP LINE OF NORMAL DSP
560      FAFE      . . .      APBUF EQU TOPSAV-2 ;POINTER TO CURRENT BUFFER
561      FAF C      . . .      DSPFLD EQU APBUF-2 ;POINTER TO CURRENT FIELD
562      FAFB      . . .      MODSAV EQU DSPFLD-1 ;SAVE HECJK BITS
563      6079      . . .      ;*****
564      6079      . . .      ; AXIS AND TIC VARIABLES
565      6079      . . .      ;*****
566      FAF9      . . .      XAX EQU MODSAV-2 ;X AXIS LOCATION
567      FAF7      . . .      YAX EQU XAX-2 ;Y AXIS LOCATION
568      FAF3      . . .      CURTIC EQU YAX-4 ;CURRENT TIC VALUE
569      FAF2      . . .      LBLCNT EQU CURTIC-1 ;COUNTER FOR TIC LABELS
570      FAF0      . . .      LBLPTR EQU LBLCNT-2 ;POINTER TO NEW TIC LABEL
571      FAEF      . . .      CHRCNT EQU LBLPTR-1 ;NO. OF CHARS IN TIC LABEL
572      FAEE      . . .      ECNT EQU CHRCNT-1 ;LOCATION OF E IN TIC LABEL
573      FAEC      . . .      TICPTR EQU ECNT-2 ;POINTER TO TIC SPACING
574      FAE8      . . .      FPSAVE EQU TICPTR-4 ;FP SCRATCH AREA
575      FAE4      . . .      FPSAV2 EQU FPSAVE-4 ;FLOATING POINT SCRATCH
576      FAE3      . . .      TICFLG EQU FPSAV2-1 ;TIC FLAG
577      FAE3      . . .      APLAST EQU TICFLG
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 17
=====
579      6079      . . .      ;*****
580      6079      . . .      ; MORE GRAPHICS STORAGE
581      6079      . . .      ;*****
582      FAE2      . . .      GINCHR EQU  APLAST-1 ;GRAPHICS INPUT CHAR
583      FAE0      . . .      M360M2 EQU  GINCHR-2 ;-360/ZOOM SIZE
584      FADE      . . .      P180M2 EQU  M360M2-2 ;+180/ZOOM SIZE
585      FADD      . . .      PTR1 EQU    P180M2-1 ;PRINTER FLAGS
586      FADC      . . .      PTR2 EQU    PTR1-1
587      FADB      . . .      LNTYPE EQU  PTR2-1 ;CURRENT LINE TYPE
588      FADB      . . .      LAST EQU   LNTYPE
589      6079      . . .      ;*****
590      6079      . . .      ; START OF MENU IN DISPLAY MEMORY
591      6079      . . .      ;*****
592      FADA      . . .      MUBUF EQU  LAST-1
593      6079      . . .      ;
594      6079      . . .      ;
595      6079      . . .      ;*****
596      6079      . . .      ; AUTO PLOT EQUATES
597      6079      . . .      ;*****
598      0004      . . .      XMINFD EQU  4 ;MIN X FIELD
599      0005      . . .      XMAXFD EQU  5 ;MAX X FIELD
600      0006      . . .      YMINFD EQU  6 ;MIN Y FIELD
601      0007      . . .      YMAXFD EQU  7 ;MAX Y FIELD
602      0008      . . .      XLBLFD EQU  8 ;X LABEL SPACING FIELD
603      0009      . . .      XTICFD EQU  9 ;X TIC SPACING FIELD
604      000A      . . .      YLBLFD EQU 10 ;Y LABEL SPACING FIELD
605      000B      . . .      YTICFD EQU 11 ;Y TIC SPACING FIELD
606      000C      . . .      SKPFLD EQU 12 ;LINES TO SKIP FIELD
607      000D      . . .      CNTFLD EQU 13 ;POINTS COUNT FIELD
608      000E      . . .      GRIDFD EQU 14 ;WANT GRID FIELD
609      000F      . . .      FRMFLD EQU 15 ;PLOT FROM DISPLAY FIELD
610      0000      . . .      FLD1 EQU   0 ;FIRST FIELD
611      000F      . . .      BOTFLD EQU 15 ;LAST FIELD IN MENU
612      0014      . . .      MAXLEN EQU 20 ;MAX ASCII NUMBER LENGTH
613      0046      . . .      XOFSET EQU 70 ;X FRAME OFFSET
614      002D      . . .      YOFSET EQU 45 ;Y FRAME OFFSET
615      0002      . . .      MINLEN EQU  2 ;MINOR TIC LENGTH
616      0004      . . .      LBLEN EQU   4 ;LABELED TIC LENGTH
617      0266      . . .      XAXLEN EQU 614 ;X FRAME LENGTH
618      0130      . . .      YAXLEN EQU 304 ;Y FRAME LENGTH
619      0001      . . .      RSTJMP EQU  1 ;RESTART 1 JUMP
620      0088      . . .      GRDPAT EQU 210Q ;PATTERN FOR GRID LINES
621      0010      . . .      MAXAP EQU  16 ;MAX AUTO PLOT STRING LEN
622      0008      . . .      APOFST EQU 10Q ;CONVERT FROM LETTER TO FIEL
623      004F      . . .      MAXCOL EQU  79 ;MAX SCREEN COLUMN
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  18
=====
625      6079      . . .      ;*****
626      6079      . . .      ; APFLGS SYMBOLS
627      6079      . . .      ;*****
628      0001      . . .      NIP      EQU  1Q      ;NUMBER IN PROGRESS
629      0002      . . .      APIP     EQU  2Q      ;AUTO PLOT IN PROGRESS
630      0004      . . .      HAVEE   EQU  4Q      ;FOUND E IN STRING
631      0008      . . .      HAVEP   EQU 10Q      ;FOUND . IN STRING
632      0010      . . .      HAVED   EQU 20Q      ;FOUND DIGIT IN STRING
633      0020      . . .      NMBD    EQU 40Q      ;NEXT CHAR MUST BE DIGIT
634      0040      . . .      HAVES   EQU 100Q     ;FOUND + OR - IN STRING
635      0080      . . .      NEXTRM  EQU 200Q     ;NEXT CHARACTER TERMINATES N
636      6079      . . .      ;*****
637      6079      . . .      ;APFLG2 SYMBOLS
638      6079      . . .      ;*****
639      0001      . . .      MENUON  EQU  1Q      ;AUTOPLT MENU ON
640      0002      . . .      APDISP  EQU  2Q      ;AUTOPLT FROM DISPLAY MEMOR
641      0004      . . .      HAVEX   EQU  4Q      ;HAVE X DATA COLUMN
642      0008      . . .      HAVEY   EQU 10Q      ;HAVE Y DATA COLUMN
643      0020      . . .      DECPNT  EQU 40Q      ;WANT DECIMAL POINT IN TIC
644      0080      . . .      TICLBL  EQU 200Q     ;WANT LABELED TICS
645      0010      . . .      WANTAP  EQU  20Q      ;WANT AUTOPLT ON
646      0040      . . .      WANTAX  EQU 100Q     ;WANT AXES DRAWN
647      6079      . . .      ; EQUATES TO PARAMETER BUFFERS
648      FBC3      . . .      NMCLBF  EQU  APB1
649      FBC2      . . .      XCOLBF  EQU  APB2
650      FBC1      . . .      YCOLBF  EQU  APB3
651      FBC0      . . .      LINEBF  EQU  APB4
652      FBBC      . . .      XMINBF  EQU  APB5
653      FBB8      . . .      XMAXBF  EQU  APB6
654      FBB4      . . .      YMINBF  EQU  APB7
655      FBB0      . . .      YMAXBF  EQU  APB8
656      FBAC      . . .      XLBLBF  EQU  APB9
657      FBA8      . . .      XTICBF  EQU  APB10
658      FBA4      . . .      YLBLBF  EQU  APB11
659      FBA0      . . .      YTICBF  EQU  APB12
660      FB9E      . . .      SKPBF   EQU  APB13
661      FB9C      . . .      CNTBF   EQU  APB14
662      FB9A      . . .      GRIDBF  EQU  APB15
663      FB98      . . .      FROMBF  EQU  APB16
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 19
665	6079	.	.	;*****	
666	6079	.	.	; RANGE TABLES FOR GRAPHICS	
667	6079	.	.	; THE ADDRESS FOR AN INDEX TABLE MUST BE BELOW	
668	6079	.	.	; 32K SO THAT THE MSB ISNT SET. CHINT HAS BEEN	
669	6079	.	.	; MODIFIED TO ALLOW A JUMP ADDRESS TO BE	
670	6079	.	.	; GREATER THAN 32K IF THE MSB OF THE UPPER BOUND	
671	6079	.	.	; IS SET (THE MSB OF THE JUMP ADDRESS IS SET TO	
672	6079	.	.	; THE MSB OF THE UPPER BOUND)	
673	6079	.	.	;*****	
674	6079	.	.	;THESE CONSTANTS SHIFT THE MSB OF AN ADDRESS FIRST	
675	6079	.	.	;INTO BIT 0, THEN INTO BIT 7, SO THAT THE MSB OF	
676	6079	.	.	;THE UPPER BOUND CAN BE SET ACCORDING TO THE ADDR	
677	8000	.	.	XDIV EQU 100000 ;LEFT SHIFT 15	
678	0080	.	.	XMUL EQU 200 ;RIGHT SHIFT 7	
679	6079	.	.	;*****	
680	6079	.	.	; GTAB--ESC * HAS BEEN RECEIVED, NEXT CHAR	
681	6079	.	.	; DETERMINES TYPE OF ESCAPE SEQUENCE	
682	6079	.	.	;*****	
683	6076	.	.	GTAB EQU \$-3	
684	6079	70	70	DB 160Q,PLTSEQ/XDIV*XMUL+160Q ;SMALL P	
685	607B	91	E3	DW PLTSEQ+B15 ;PLOTING	
686	607D	.	.	;	
687	607D	64	64	DB 144Q,DSPSEQ/XDIV*XMUL+144Q ;SMALL D	
688	607F	6C	ED	DW DSPSEQ+B15 ;DISPLAY CONTROL	
689	6081	.	.	;	
690	6081	6D	6D	DB 155Q,MODSEQ/XDIV*XMUL+155Q ;SMALL M	
691	6083	0A	F1	DW MODSEQ+B15 ;DRAWING MODE	
692	6085	.	.	;	
693	6085	74	74	DB 164Q,TEKSEQ/XDIV*XMUL+164Q ;SMALL T	
694	6087	D1	E8	DW TEKSEQ+B15 ;TEK MODE	
695	6089	.	.	;	
696	6089	61	E1	DB 141Q,APSEQ/XDIV*XMUL+141Q ;SMALL A	
697	608B	8F	29	DW APSEQ+B15 ;AUTO PLOT	
698	608D	.	.	;	
699	608D	6C	6C	DB 154Q,LBLSEQ/XDIV*XMUL+154Q ;SMALL L	
700	608F	6A	EC	DW LBLSEQ+B15 ;LABEL	
701	6091	.	.	;	
702	6091	73	73	DB 163Q,STATUS/XDIV*XMUL+163Q ;SMALL S	
703	6093	58	F3	DW STATUS+B15	
704	6095	.	.	;	
705	6095	72	F2	DB 162Q,IGNSEQ/XDIV*XMUL+162Q ;SMALL R	
706	6097	DC	19	DW IGNSEQ+B15 ;IGNORE FOR NOW	
707	6099	.	.	;	
708	6099	62	E2	DB 142Q,IGNSEQ/XDIV*XMUL+142Q ;SMALL B	
709	609B	DC	19	DW IGNSEQ+B15 ;IGNORE FOR NOW	
710	609D	.	.	;	
711	609D	60	FE	DB 140Q,IGNSEQ/XDIV*XMUL+176Q ;LOWER CASE	
712	609F	DC	19	DW IGNSEQ+B15	
713	60A1	.	.	;	
714	60A1	00	7F	DB 0Q,177Q ;ANYTHING ELSE	

13255
2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                PAGE 20
=====
  715     60A3     4F  80  .                DW  ZESCND+B15                ;TERMINATE
=====
```


				=====		PAGE 21	
ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS				
				=====			
717	60A5	.	.	.	;*****		
718	60A5	.	.	.	; PLTTAB--USED IN VECTOR PLOTTING SEQUENCE		
719	60A5	.	.	.	;*****		
720	60A2	.	.	.	PLTTAB EQU	\$-3	
721	60A5	20	3F	.	DB	40Q,PLTPRM/XDIV*XMUL+77Q	;PARAMETER
722	60A7	AE	E3	.	DW	PLTPRM+B15	
723	60A9	61	6C	.	DB	141Q,154Q	;SMALL A-L
724	60AB	BD	60	.	DW	PINDX	;USE INDEX
725	60AD	41	49	.	DB	101Q,111Q	;CAP A-L
726	60AF	BD	60	.	DW	PINDX	;USE INDEX
727	60B1	40	7F	.	DB	100Q,NOP1/XDIV*XMUL+177Q	;ANY OTHER
728	60B3	49	E4	.	DW	NOP1+B15	;LETTER
729	60B5	1B	1B	.	DB	33Q,PLTESC/XDIV*XMUL+33Q	;ESCAPE
730	60B7	4F	E4	.	DW	PLTESC+B15	
731	60B9	00	9F	.	DB	0Q,NOFUNC/XDIV*XMUL+37Q	;ANY OTHER
732	60B8	DB	19	.	DW	NOFUNC+B15	;CONTROL COD
733	60BD	.	.	.	;		
734	60BD	.	.	.	PINDX EQU	\$	
735	60BD	01	64	.	DW	PENUP	;A--RAISE PEN
736	60BF	F6	63	.	DW	PENDN	;B--LOWER PEN
737	60C1	0C	64	.	DW	USEGC	;C--USE GC POSITION AS NEW P
738	60C3	21	64	.	DW	ONEDOT	;D--SINGLE POINT,LIFT PEN
739	60C5	43	64	.	DW	PNORG1	;E--SET RELOC ORG = CUR PT.
740	60C7	B2	63	.	DW	ASABFT	;F--USE ASCII ABSOLUTE FORMA
741	60C9	B8	63	.	DW	ASINFT	;G--USE ASCII INCREM. FORMAT
742	60CB	BE	63	.	DW	ASRLFT	;H--USE ASCII RELOC FORMAT
743	60CD	D1	63	.	DW	ABSFMT	;I--USE ABSOLUTE FORMAT
744	60CF	D7	63	.	DW	SHTFMT	;J--USE SHORT INCR. FORMAT
745	60D1	DD	63	.	DW	INCFMT	;K--USE INCREMENTAL FORMAT
746	60D3	E3	63	.	DW	RELFMT	;L--USE RELOCATABLE FORMAT

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 22
=====
748      60D5      . . .      ;*****
749      60D5      . . .      ; DSPTAB--USED BY DISPLAY CONTROL SEQUENCE
750      60D5      . . .      ;*****
751      60D2      . . .      DSPTAB EQU $-3
752      60D5      20 3F .      DB      40Q,GPARAM/XDIV*XMUL+77Q ;PARAMETER
753      60D7      A6 EC .      DW      GPARAM+B15
754      60D9      61 74 .      DB      141Q,164Q ;SMALL A-T
755      60DB      ED 60 .      DW      DINDX ;USE INDEX
756      60DD      41 54 .      DB      101Q,124Q ;CAP A-T
757      60DF      ED 60 .      DW      DINDX ;USE INDEX
758      60E1      40 FF .      DB      100Q,NOP/XDIV*XMUL+177Q ;ANY OTHER
759      60E3      C1 19 .      DW      NOP+B15 ;LETTER
760      60E5      1B 1B .      DB      33Q,33Q ;ESCAPE
761      60E7      B7 80 .      DW      ZESCAP+B15
762      60E9      00 9F .      DB      0Q,NOFUNC/XDIV*XMUL+37Q ;ANY OTHER
763      60EB      DB 19 .      DW      NOFUNC+B15 ;CONTROL COD
764      60ED      . . .      ;
765      60ED      . . .      DINDX EQU $
766      60ED      72 6D .      DW      GCLEAR ;A--CLEAR THE GRAPHICS SCREE
767      60EF      7D 6D .      DW      GSET ;B--SET THE GRAPHICS SCREEN
768      60F1      A6 6D .      DW      GVON ;C--ENABLE GRAPHICS VIDEO
769      60F3      E7 6D .      DW      GVOFF ;D--INHIBIT GRAPHICS VIDEO
770      60F5      12 6E .      DW      ANVON ;E--ENABLE A/N VIDEO
771      60F7      23 6E .      DW      ANVOFF ;F--INHIBIT A/N VIDEO
772      60F9      3B 6E .      DW      ZON ;G--TURN ZOOM ON
773      60FB      58 6E .      DW      ZOFF ;H--TURN ZOOM OFF
774      60FD      89 6E .      DW      ZSIZE ;I--SET ZOOM SIZE
775      60FF      68 6F .      DW      ZPOS ;J--SET ZOOM POSITION
776      6101      D0 6F .      DW      TGCON ;K--TURN GRAPHICS CURSOR ON
777      6103      F6 6F .      DW      TGCOFF ;L--TURN GRAPHICS CURSOR OFF
778      6105      31 70 .      DW      TRBON ;M--TURN RUBBER BAND LINE ON
779      6107      48 70 .      DW      TRBOFF ;N--TURN RUBBER BAND LINE OF
780      6109      81 6F .      DW      AGCPOS ;O--SET CURSOR POSITION ABS
781      610B      A6 6F .      DW      IGCPOS ;P--SET CURSOR POS INCREMETA
782      610D      13 70 .      DW      ACON ;Q--TURN A/N CURSOR ON
783      610F      24 70 .      DW      ACOFF ;R--TURN A/N CURSOR OFF
784      6111      F1 75 .      DW      GTXON ;S--TURN GRAPHICS TEXT ON
785      6113      18 76 .      DW      GTXOF ;T--TURN GRAPHICS TEXT OFF
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
787	6115	.	.	*****
788	6115	.	.	; MODTAB--USED BY DRAWING MODE SEQUENCE
789	6115	.	.	*****
790	6112	.	.	MODTAB EQU S-3
791	6115	20	3F	DB 40Q,GPARAM/XDIV*XMUL+77Q ;PARAMETER
792	6117	A6	EC	DW GPARAM+B15
793	6119	61	72	DB 141Q,162Q ;SMALL A-R
794	6118	2D	61	DW MINDX ;USE INDEX
795	611D	41	52	DB 101Q,122Q ;CAP A-R
796	611F	2D	61	DW MINDX ;USE INDEX
797	6121	40	FF	DB 100Q,NOP/XDIV*XMUL+177Q ;ANY OTHER
798	6123	C1	19	DW NOP+B15 ;LETTER
799	6125	1B	1B	DB 33Q,33Q ;ESCAPE
800	6127	B7	80	DW ZESCAP+B15
801	6129	00	9F	DB 0Q,NOFUNC/XDIV*XMUL+37Q ;ANY OTHER
802	612B	DB	19	DW NOFUNC+B15 ;CONTROL COD
803	612D	.	.	;
804	612D	.	.	MINDX EQU \$
805	612D	15	72	DW SETMOD ;A--SET DRAWING MODE
806	612F	39	72	DW SETLIN ;B--SET LINE TYPE
807	6131	AD	72	DW DEFLP ;C--DEFINE LINE PATTERN
808	6133	EB	72	DW DEFAP ;D--DEFINE AREA PATTERN
809	6135	10	71	DW ABFILL ;E--ABSOLUTE AREA FILL
810	6137	D9	71	DW RLFILL ;F--RELOCATABLE AREA FILL
811	6139	C1	99	DW NOP ;G--NOT USED
812	613B	C1	99	DW NOP ;H--NOT USED
813	613D	C1	99	DW NOP ;I--NOT USED
814	613F	2B	73	DW SETORG ;J--SET RELOCATABLE ORIGIN
815	6141	44	73	DW PENORG ;K--SET ORIGIN=CURRENT PEN
816	6143	4E	73	DW GORG ;L--SET ORIGIN=GC POSITION
817	6145	DC	77	DW TXSIZE ;M--SET GRAPHICS TEXT SIZE
818	6147	02	78	DW TXANGL ;N--SET GRAPHICS TEXT ANGLE
819	6149	26	76	DW SLNTOF ;O--TURN TEXT SLANT ON
820	614B	3A	76	DW SLNTOF ;P--TURN SLANT OFF
821	614D	44	99	DW LORG ;Q--SET LABEL ORIGIN
822	614F	7F	99	DW DEFAULT ;R--SET DEFAULTS

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 24
824	6151	.	.	*****	
825	6151	.	.	; TEK MODE RANGE TABLES	
826	6151	.	.	*****	
827	614E	.	.	TEKTAB EQU \$-3 ;ESC * T RECEIVED	
828	6151	61	63	DB 141Q,143Q ;SMALL A-C	
829	6153	69	61	DW TKINDX ;USE INDEX	
830	6155	41	43	DB 101Q,103Q ;CAP A-C	
831	6157	69	61	DW TKINDX ;USE INDEX	
832	6159	20	3F	DB 40Q,GPARAM/XDIV*XMUL+77Q ;PARAMETER	
833	6158	A6	EC	DW GPARAM+B15	
834	615D	40	FF	DB 100Q,NOP/XDIV*XMUL+177Q ;ANY OTHER	
835	615F	C1	19	DW NOP+B15 ;LETTER	
836	6161	1B	1B	DB 33Q,33Q ;ESCAPE	
837	6163	B7	80	DW ZESCAP+B15 ;SET UP FOR NEW SEQUENCE	
838	6165	00	FF	DB 0Q,NOFUNC/XDIV*XMUL+177Q ;ANYTHIN ELS	
839	6167	DB	19	DW NOFUNC+B15	
840	6169	.	.	;	
841	6169	.	.	TKINDX EQU \$	
842	6169	00	69	DW TRMSTP ;A--SET GIN TERMINATOR	
843	616B	0E	69	DW SETBRK ;B--SET PAGE FULL BREAK FLAG	
844	616D	2D	69	DW SETBSY ;C--SET PAGE FULL BUSY FLAG	
845	616F	.	.	;	
846	616C	.	.	TKGSTB EQU \$-3 ;GS RECEIVED	
847	616F	20	3F	DB 40Q,HIXY/XDIV*XMUL+77Q ;HI X OR Y	
848	6171	87	E9	DW HIXY+B15	
849	6173	40	5F	DB 100Q,LOWX/XDIV*XMUL+137Q ;LOW X BYTE	
850	6175	AE	E9	DW LOWX+B15	
851	6177	60	7F	DB 140Q,LOWY/XDIV*XMUL+177Q ;LOW Y BYTE	
852	6179	A1	E9	DW LOWY+B15	
853	617B	1D	1D	DB 35Q,TKSTUP/XDIV*XMUL+35Q ;GS	
854	617D	62	E9	DW TKSTUP+B15	
855	617F	1F	1F	DB 37Q,GSEND/XDIV*XMUL+37Q ;US	
856	6181	7E	E9	DW GSEND+B15	
857	6183	0D	0D	DB 15Q,TEKCR/XDIV*XMUL+15Q ;CR	
858	6185	47	EA	DW TEKCR+B15	
859	6187	1B	1B	DB 33Q,TEKESC/XDIV*XMUL+33Q ;ESC	
860	6189	17	EA	DW TEKESC+B15	
861	618B	00	FF	DB 0Q,NOFUNC/XDIV*XMUL+177Q ;ANYTHIN ELS	
862	618D	DB	19	DW NOFUNC+B15	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
864	618C	.	.	TESCTB EQU \$-3 ;ESC RECEIVED IN TEK GRAPHIC
865	618F	0C	0C	DB 14Q,PAGE/XDIV*XMUL+14Q ;FF
866	6191	1D	EA	DW PAGE+B15
867	6193	1A	1A	DB 32Q,STGIN/XDIV*XMUL+32Q ;SUB
868	6195	DE	EA	DW STGIN+B15
869	6197	05	05	DB 5Q,TEKCP/XDIV*XMUL+5Q ;ENQ
870	6199	B1	EA	DW TEKCP+B15
871	619B	17	17	DB 27Q,TEKHC/XDIV*XMUL+27Q ;ETB
872	619D	57	EC	DW TEKHC+B15
873	619F	00	7F	DB 0Q,TEKRPT/XDIV*XMUL+177Q ;ANYTHIN ELS
874	61A1	34	EA	DW TEKRPT+B15
875	61A3	.	.	;
876	61A3	.	.	;
877	61A0	.	.	GINTAB EQU \$-3
878	61A3	0D	0D	DB 15Q,GINCR/XDIV*XMUL+15Q ;CR
879	61A5	9A	EB	DW GINCR+B15
880	61A7	1B	1B	DB 33Q,GINESC/XDIV*XMUL+33Q ;ESC
881	61A9	69	EB	DW GINESC+B15
882	61AB	00	7F	DB 0Q,GINCH/XDIV*XMUL+177Q ;ANYTHIN ELS
883	61AD	14	EB	DW GINCH+B15
884	61AF	.	.	;
885	61AF	.	.	;
886	61AC	.	.	GNECTB EQU \$-3
887	61AF	00	7F	DB 0Q,ESCCH/XDIV*XMUL+177Q ;ANY CHAR
888	61B1	81	EB	DW ESCCH+B15
889	61B3	.	.	;
890	61B3	.	.	;
891	61B3	.	.	;
892	61B0	.	.	IGNTAB EQU \$-3
893	61B3	1B	1B	DB 33Q,33Q ;ESCAPE
894	61B5	B7	80	DW ZESCAP+B15 ;EXECUTE
895	61B7	40	5F	DB 100Q,137Q ;UPPER CASE
896	61B9	4F	80	DW ZESCND+B15
897	61BB	00	FF	DB 0Q,NOFUNC/XDIV*XMUL+177Q ;ANY OTHER
898	61BD	DB	19	DW NOFUNC+B15

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 26
=====
  900     61BF      . . .      ;*****
  901     61BF      . . .      ; LABELS RANGE TABLES
  902     61BF      . . .      ;*****
  903     61BC      . . .      LBLTAB EQU $-3
  904     61BF      20 7F .      DB 40Q,PUTCHR/XDIV*XMUL+177Q ;CHAR
  905     61C1      3B F8 .      DW PUTCHR+B15
  906     61C3      0D 0D .      DB 15Q,LBLCR/XDIV*XMUL+15Q ;CR
  907     61C5      79 EC .      DW LBLCR+B15
  908     61C7      0A 0A .      DB 12Q,LBLLF/XDIV*XMUL+12Q ;LF
  909     61C9      82 EC .      DW LBLLF+B15
  910     61CB      1B 1B .      DB 33Q,LBLESC/XDIV*XMUL+33Q ;ESCAPE
  911     61CD      97 EC .      DW LBLESC+B15
  912     61CF      00 FF .      DB 0Q,NOFUNC/XDIV*XMUL+177Q ;ANY OTHER
  913     61D1      0B 19 .      DW NOFUNC+B15
  914     61D3      . . .      ;
  915     61D0      . . .      LBLTB2 EQU $-3
  916     61D3      0D 0D .      DB 15Q,LBLCR2/XDIV*XMUL+15Q ;CR
  917     61D5      88 EC .      DW LBLCR2+B15
  918     61D7      0A 0A .      DB 12Q,LBLLF2/XDIV*XMUL+12Q ;LF
  919     61D9      91 EC .      DW LBLLF2+B15
  920     61DB      00 7F .      DB 0Q,LBLEND/XDIV*XMUL+177Q ;ANY OTHER
  921     61DD      E1 F5 .      DW LBLEND+B15
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 27
=====
 923     61DF      .      .      .      ;*****
 924     61DF      .      .      .      ; STATUS RANGE TABLES
 925     61DF      .      .      .      ;*****
 926     61DF      .      .      .      ;
 927     61DC      .      .      .      STATTB EQU $-3
 928     61DF      20     3F      .      DB      40Q,GPARAM/XDIV*XMUL+77Q      ;PARAMETER
 929     61E1      A6     EC      .      DW      GPARAM+B15
 930     61E3      40     SF      .      DB      100Q,GSTAT/XDIV*XMUL+137Q     ;ANY CAP
 931     61E5      61     F3      .      DW      GSTAT+B15
 932     61E7      60     FF      .      DB      140Q,NOP/XDIV*XMUL+177Q     ;LOWER CASE
 933     61E9      C1     19      .      DW      NOP+B15
 934     61EB      1B     1B      .      DB      33Q,33Q      ;ESCAPE
 935     61ED      B7     80      .      DW      ZESCAP+B15
 936     61EF      00     FF      .      DB      0Q,NOFUNC/XDIV*XMUL+177Q     ;ANY OTHER
 937     61F1      DB     19      .      DW      NOFUNC+B15
 938     61F3      .      .      .      ;
 939     61F0      .      .      .      GCWTAB EQU $-3
 940     61F3      1B     1B      .      DB      33Q,GCWESC/XDIV*XMUL+33Q     ;ESCAPE
 941     61F5      1C     F4      .      DW      GCWESC+B15
 942     61F7      00     7F      .      DB      0Q,GCWESC/XDIV*XMUL+177Q     ;ANY OTHER
 943     61F9      11     F4      .      DW      GCWCHR+B15
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE  28
=====
 945      61FB      . . .      ;*****
 946      61FB      . . .      ; AUTO PLOT RANGE TABLES
 947      61FB      . . .      ;*****
 948      61F8      . . .      MUTB EQU $-3 ;MAIN MENU TABLE
 949      61FB      30 B9 .      DB 60Q,ADDCHR/XDIV*XMUL+71Q ;0-9
 950      61FD      A4 2E .      DW ADDCHR+B15
 951      61FF      20 A0 .      DB 40Q,ADDCHR/XDIV*XMUL+40Q ;SPACE
 952      6201      A4 2E .      DW ADDCHR+B15
 953      6203      2B AB .      DB 53Q,ADDCHR/XDIV*XMUL+53Q ;+
 954      6205      A4 2E .      DW ADDCHR+B15
 955      6207      2D AD .      DB 55Q,ADDCHR/XDIV*XMUL+55Q ;-
 956      6209      A4 2E .      DW ADDCHR+B15
 957      620B      2E AE .      DB 56Q,ADDCHR/XDIV*XMUL+56Q ;.
 958      620D      A4 2E .      DW ADDCHR+B15
 959      620F      45 C5 .      DB 105Q,ADDCHR/XDIV*XMUL+105Q ;CAP E
 960      6211      A4 2E .      DW ADDCHR+B15
 961      6213      65 E5 .      DB 145Q,ADDE/XDIV*XMUL+145Q ;SMALL E
 962      6215      C2 2E .      DW ADDE+B15
 963      6217      08 0D .      DB 10Q,15Q ;CONTROL CODES
 964      6219      27 62 .      DW MUI ;USE INDEX
 965      621B      1B 9B .      DB 33Q,MUESC/XDIV*XMUL+33Q ;ESCAPE
 966      621D      89 2D .      DW MUESC+B15
 967      621F      00 1F .      DB 0Q,37Q ;CONTROL CODES
 968      6221      F0 80 .      DW ZCKCTL+B15 ;CHECK FOR BLOCK TRIGGER
 969      6223      00 FF .      DB 0Q,CHKCH/XDIV*XMUL+177Q ;ANYTHIN ELS
 970      6225      19 39 .      DW CHKCH+B15
 971      6227      . . .      ;
 972      6227      . . .      MUI EQU $
 973      6227      E6 AD .      DW MOVLFT ;BS--LEFT ONE COL
 974      6229      C5 AD .      DW MOVDN ;TAB--DOWN ONE FIELD
 975      622B      C5 AD .      DW MOVDN ;LINE FEED
 976      622D      DB 99 .      DW NOFUNC ;VERT TAB--IGNORE
 977      622F      DB 99 .      DW NOFUNC ;FORM FEED--IGNORE
 978      6231      BB AD .      DW MOVST ;RETURN-MOVE TO ST OF FIELD
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
						29
980	6230	.	.	METB	EQU \$-3 ;ESCAPE RECEIVED IN MENU	
981	6233	41	44	.	DB 101Q,104Q ;A-D	
982	6235	5B	62	.	DW MEI ;USE INDEX	
983	6237	48	C8	.	DB 110Q,HOME/XDIV*XMUL+110Q ;CAP H	
984	6239	9B	2D	.	DW HOME+B15	
985	623B	68	E8	.	DB 150Q,HOME/XDIV*XMUL+150Q ;SMALL H	
986	623D	9B	2D	.	DW HOME+B15	
987	623F	4A	CB	.	DB 112Q,CLRFLD/XDIV*XMUL+113Q ;CAP J OR	
988	6241	AB	2D	.	DW CLRFLD+B15	
989	6243	46	C6	.	DB 106Q,HOMEDN/XDIV*XMUL+106Q ;CAP F	
990	6245	A3	2D	.	DW HOMEDN+B15	
991	6247	69	E9	.	DB 151Q,MOVUP/XDIV*XMUL+151Q ;SMALL I	
992	6249	D2	2D	.	DW MOVUP+B15	
993	624B	2A	AA	.	DB 52Q,APGSEQ/XDIV*XMUL+52Q ;ASTERISK	
994	624D	95	2D	.	DW APGSEQ+B15	
995	624F	26	A6	.	DB 46Q,PRMABT/XDIV*XMUL+46Q ;AMPERSAND	
996	6251	47	2A	.	DW PRMABT+B15	
997	6253	64	64	.	DB 144Q,144Q ;SMALL D	
998	6255	E7	80	.	DW ZENTER+B15	
999	6257	00	7F	.	DB 0Q,177Q ;ANY THING ELSE	
1000	6259	4F	80	.	DW ZESCND+B15	
1001	625B	.	.	.	; MEI	
1002	625B	.	.	.	EQU \$	
1003	625B	D2	AD	.	DW MOVUP ;A--CURSOR UP ONE FIELD	
1004	625D	C5	AD	.	DW MOVDN ;B--DOWN ONE FIELD	
1005	625F	DF	AD	.	DW MOVRT ;C--MOVE RIGHT ONE	
1006	6261	E6	AD	.	DW MOVLFT ;D--LEFT ONE FIELD	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 30
=====
1008      6263      . . .      ;
1009      6263      . . .      ;
1010      6260      . . .      APGTAB EQU $-3
1011      6263      61 E1 .      DB 141Q,APSEQ/XDIV*XMUL+141Q ;SMALL A
1012      6265      8F 29 .      DW APSEQ+B15
1013      6267      00 FF .      DB 0Q,MUABT/XDIV*XMUL+177Q ;ANY OTHER
1014      6269      3E 2A .      DW MUABT+B15
1015      6268      . . .      ;
1016      6268      . . .      ;
1017      6268      . . .      APTAB EQU $-3 ;TABLE FOR ESC * A SEQ
1018      6268      30 89 .      DB 60Q,PUTBUF/XDIV*XMUL+71Q ;0-9
1019      6260      AC 29 .      DW PUTBUF+B15
1020      626F      2B AB .      DB 53Q,PUTBUF/XDIV*XMUL+53Q ;+
1021      6271      AC 29 .      DW PUTBUF+B15
1022      6273      2D AD .      DB 55Q,PUTBUF/XDIV*XMUL+55Q ;-
1023      6275      AC 29 .      DW PUTBUF+B15
1024      6277      2E AE .      DB 56Q,PUTBUF/XDIV*XMUL+56Q ;.
1025      6279      AC 29 .      DW PUTBUF+B15
1026      6278      45 C5 .      DB 105Q,PUTBUF/XDIV*XMUL+105Q ;E
1027      627D      AC 29 .      DW PUTBUF+B15
1028      627F      68 F7 .      DB 150Q,XFRBUF/XDIV*XMUL+167Q ;SMALL H-W
1029      6281      C0 29 .      DW XFRBUF+B15
1030      6283      48 D7 .      DB 110Q,XFRBUF/XDIV*XMUL+127Q ;CAP H-W
1031      6285      C0 29 .      DW XFRBUF+B15
1032      6287      61 67 .      DB 141Q,147Q ;SMALL A-G
1033      6289      9B 62 .      DW APNDX ;USE INDEX
1034      6288      41 47 .      DB 101Q,107Q ;CAP A-G
1035      628D      9B 62 .      DW APNDX ;USE INDEX
1036      628F      40 FF .      DB 100Q,APEXIT/XDIV*XMUL+177Q ;ANY LETTER
1037      6291      4D 2A .      DW APEXIT+B15
1038      6293      1B 9B .      DB 33Q,APESC/XDIV*XMUL+33Q ;ESCAPE
1039      6295      35 2A .      DW APESC+B15
1040      6297      00 FF .      DB 0Q,NOFUNC/XDIV*XMUL+177Q ;ANYTHING ELS
1041      6299      DB 19 .      DW NOFUNC+B15
1042      6298      . . .      ;
1043      6298      . . .      APNDX EQU $
1044      6298      E8 A9 .      DW GOAP ;A--START AUTO PLOT
1045      629D      F0 A9 .      DW STOPAP ;B--STOP AUTO PLOT
1046      629F      F8 A9 .      DW DWAXES ;C--DRAW AXES
1047      62A1      00 AA .      DW CLRMNU ;D--CLEAR MENU
1048      62A3      4D AA .      DW APEXIT ;E--IGNORE
1049      62A5      1D AA .      DW DSPMNU ;F--TURN MENU ON
1050      62A7      29 AA .      DW OFFMNU ;G--TURN MENU OFF
1051      62A9      . . .      ;
1052      62A9      . . .      ;
1053      62A9      . . .      ;IGNORE CR/LF RANGE TABLE
1054      62A6      . . .      CRLFTB EQU $-3
1055      62A9      0D 0D .      DB 15Q,NOCR/XDIV*XMUL+15Q ;CR
1056      62AB      D1 F5 .      DW NOCR+B15
1057      62AD      0A 0A .      DB 12Q,NOLF/XDIV*XMUL+12Q ;LF
=====

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 31
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
1058	62AF	DA F5 .	DW NOLF+B15	31
1059	62B1	13 13 .	DB 23Q,NODC3/XDIV*XMUL+23Q ;DC3	31
1060	62B3	CB F5 .	DW NODC3+B15	31
1061	62B5	00 7F .	DB 0Q,CRLFON/XDIV*XMUL+177Q ;ANY OTHER	31
1062	62B7	E4 F5 .	DW CRLFON+B15	31

```
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 32
1064	62B9	.	.	;*****	
1065	62B9	.	.	; HRDRST --DO HARD RESET. INITIALIZE ALL GRAPHICS	
1066	62B9	.	.	; PARAMETERS AND FLAGS	
1067	62B9	.	.	; ENTRY--DONT CARE	
1068	62B9	.	.	; EXIT---ALL REGISTERS DESTROYED	
1069	62B9	.	.	;*****	
1070	62B9	.	.	HRDRST EQU \$	
1071	62B9	AF	.	XRA A	
1072	62BA	32	60 89	STA GRESET ;RESET CONTROLLER	
1073	62BD	32	20 89	STA HWFLGS ;CLEAR ALL HW FLAGS	
1074	62C0	CD	92 A2	CALL VRWAIT ;WAIT FOR HW TO SYNCH	
1075	62C3	CD	04 62	CALL HARD1 ;DO ALL BUT CLEAR	
1076	62C6	3E	80 .	MVI A,RESET ;SET THE RESET FLAG	
1077	62C8	CD	67 A2	CALL STFLG7	
1078	62CB	CD	32 69	CALL TKSTRP ;CHECK TEK MODE STRAPS	
1079	62CE	CD	50 6A	CALL TEKHOM ;HOME CURSOR IF TEK	
1080	62D1	C3	78 6D	JMP GCLR1 ;CLEAR THE SCREEN	
1081	62D4	.	.	HARD1 EQU \$	
1082	62D4	.	.	; CLEAR GRAPHICS FAST RAM	
1083	62D4	AF	.	XRA A	
1084	62D5	4F	.	MOV C,A	
1085	62D6	21	00 90	LXI H,FSTRM2 ;BASE OF 2ND FAST RAM	
1086	62D9	.	.	HRD010 EQU \$	
1087	62D9	77	.	MOV M,A ;CLEAR A BYTE	
1088	62DA	2C	.	INR L ;ALL DONE?	
1089	62DB	C2	D9 62	JNZ HRD010 ;NO	
1090	62DE	.	.	; CLEAR GRAPHICS SLOW RAM	
1091	62DE	21	FE FB	LXI H,ZDSPLM-1 ;START OF SLOW RAM	
1092	62E1	11	25 01	LXI D,ZDSPLM-MUBUF ;NO. OF BYTES	
1093	62E4	.	.	HRD015 EQU \$	
1094	62E4	71	.	MOV M,C ;CLEAR A BYTE	
1095	62E5	2B	.	DCX H ;UPDATE STORE ADDRESS	
1096	62E6	1B	.	DCX D ;UPDATE NO. OF BYTES LEFT	
1097	62E7	7A	.	MOV A,D ;ALL DONE?	
1098	62E8	B3	.	ORA E	
1099	62E9	C2	E4 62	JNZ HRD015 ;NO	
1100	62EC	CD	87 A2	CALL WAIT ;INSURE HW IDLE	
1101	62EF	CD	48 76	CALL ANGLE ;SET TEXT PARAMETERS	
1102	62F2	3E	08 .	MVI A,100 ;SET USE NEW WA FLAG	
1103	62F4	32	B2 90	STA GFLGS1 ;IN SOFTWARE	
1104	62F7	32	AE 90	STA GFLGS5 ;SET DRAW FIRST DOT FLAG	
1105	62FA	21	FC FF	LXI H,-4 ;DRAW 4 DOTS/SCAN LINE	
1106	62FD	22	02 89	SHLD VDC ;SEND TO HW	
1107	6300	.	.	; SAMPLE MUST BE ON TO ALLOW PATTERN SHIFTS	
1108	6300	3E	36 .	MVI A,660 ;MODE=SET,PAT ON, SAMPLE ON	
1109	6302	32	41 89	STA HCEJK ;GRAPHICS VIDEO ON	
1110	6305	32	B5 90	STA CURMOD ;SAVE AS CURRENT DRAWING MOD	
1111	6308	.	.	; SET DEFAULT LINE AND AREA PATTERNS TO ALL ON	
1112	6308	3E	FF .	MVI A,377Q ;DEFAULT PATTERN	
1113	630A	32	B4 90	STA CURPAT ;USE AS CURRENT PATTERN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 33
=====
1114     630D     32  A8  90          STA  LPAT          ;SET USER DEFINED LINE PAT
1115     6310     0E  0F  .           MVI  C,15          ;LOOP COUNTER
1116     6312     21  EF  FB          LXI  H,VAPAT       ;BASE OF BOTH PATTERN BUFFS
1117     6315     .   .   .           HRD020 EQU $
1118     6315     77  .   .           MOV  M,A           ;SEND PATTERN
1119     6316     23  .   .           INX  H
1120     6317     0D  .   .           DCR  C             ;ALL 16 STORED YET?
1121     6318     C2  15  63          JNZ  HRD020        ;NO, KEEP LOOPING
1122     631B     .   .   .           ; AUTO PLOT INITIALIZATION
1123     631B     CD  2F  BD          CALL INIT          ;INITIALIZE FLOAT POINT STUF
1124     631E     3E  00  .           MVI  A,FLD1        ;SET CURRENT FIELD TO
1125     6320     32  02  FB          STA  MUFLD         ;THE FIRST ONE
1126     6323     CD  1F  AB          CALL MUTODM        ;PUT AP MENU INTO DSP MEM
1127     6326     .   .   .           ; LIFT THE PEN
1128     6326     3E  01  .           MVI  A,MOVE
1129     6328     CD  26  A2          CALL STFLG1
1130     632B     .   .   .           ; SET NORMAL CLIPPING LIMITS
1131     632B     .   .   .           HRD1  EQU $
1132     632B     21  00  00          LXI  H,0           ;MIN VALUE IS 0
1133     632E     22  72  90          SHLD XMIN
1134     6331     22  6E  90          SHLD YMIN
1135     6334     21  31  FD          LXI  H,-719        ;STORE AS -(MAX)
1136     6337     22  70  90          SHLD XMAX          ;MAX X IS 719
1137     633A     21  99  FE          LXI  H,-359
1138     633D     22  6C  90          SHLD YMAX          ;MAXY Y IS 359
1139     6340     .   .   .           ; SET NEW BOUNDS CODE FOR CURRENT POINT
1140     6340     .   .   .           ; SINCE CLIP LIMIT HAS CHANGED
1141     6340     .   .   .           HRD2  EQU $
1142     6340     2A  DE  90          LHLD XCURR
1143     6343     22  DA  90          SHLD XNEW
1144     6346     2A  DC  90          LHLD YCURR
1145     6349     22  D8  90          SHLD YNEW
1146     634C     C3  2D  98          JMP  CPUPD1        ;UPDATE CURRENT POINT
1147     634F     .   .   .           ;*****
1148     634F     .   .   .           ; SOFT RESET
1149     634F     .   .   .           ; ENTRY--DONT CARE
1150     634F     .   .   .           ; EXIT---ALL REGISTERS DESTROYED
1151     634F     .   .   .           ;*****
1152     634F     .   .   .           SFTRST EQU $
1153     634F     3E  30  .           MVI  A,SUPCHR+GINMOD ;CLEAR ECHO SUPRESS
1154     6351     CD  53  A2          CALL CLTKFL        ;AND GIN MODE FLAGS
1155     6354     .   .   .           ; TURN LABEL OFF
1156     6354     CD  9D  6C          CALL LBLOFF
1157     6357     3E  02  .           MVI  A,APLABL     ;CLEAR AUTO PLOT LABEL IN
1158     6359     CD  6D  A2          CALL CLFLG7        ;PROGRESS
1159     635C     AF  .   .           XRA  A
1160     635D     32  DD  FA          STA  PTR1          ;CLEAR PRINTER FLAGS
1161     6360     CD  82  B8          CALL APLTUF        ;TURN AUTO PLOT OFF
1162     6363     CD  2B  63          CALL HRD1          ;RESET CLIP LIMITS
1163     6366     C3  AB  9E          JMP  ENAB0         ;RE-ENABLE THE CURSOR
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 34
=====
1165      6369      . . .      ;*****
1166      6369      . . .      ; DSPTST--TEST TO SEE IF CHARACTER SHOULD BE
1167      6369      . . .      ; PUT INTO THE A/N DISPLAY.
1168      6369      . . .      ; IF G TEXT ON, PUT CHAR INTO GRAPHICS MEMORY
1169      6369      . . .      ; ENTRY DCHAR = CHARACTER
1170      6369      . . .      ; EXIT CY => DO NOT PUT CHAR IN A/N
1171      6369      . . .      ; NC => PUT CHAR INTO A/N
1172      6369      . . .      ; IF CHAR IS PUT INTO A/N, ONLY A IS DESTROYED
1173      6369      . . .      ; IF PUT CHAR INTO GRAPHICS, ALL DESTROYED
1174      6369      . . .      ;*****
1175      6369      . . .      DSPTST EQU $
1176      6369      . . .      ; IF SOFT KEYS UP, PROCESS NORMALLY
1177      6369      CD C6 00      CALL ZCHKSF      ;ARE THEY UP?
1178      636C      C0 . .      RNZ              ;YES, PUT INTO A/N
1179      636D      CD D9 9D      CALL DFCHK       ;IN DISPLAY FUNCTIONS?
1180      6370      C0 . .      RNZ              ;YES, PUT INTO A/N
1181      6371      . . .      ; IF ALL LEDS ARE ON, SELF TEST IS IN PROGRESS
1182      6371      . . .      ; PUT TEXT IN A/N IF SO
1183      6371      CD DF 9D      CALL LEDCHK      ;ALL LEDS ON?
1184      6374      C8 . .      RZ              ;YES, PUT INTO A/N
1185      6375      . . .      ; PUT CHAR INTO GRAPHICS
1186      6375      CD 3B 78      CALL PUTCHR      ;NO, PUT INTO GRAPHICS
1187      6378      37 . .      STC              ;CY => DONT PROCESS FURTHER
1188      6379      C9 . .      RET
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
1190	637A	.	.	;*****
1191	637A	.	.	; GSETUP--ESC * HAS BEEN RECEIVED. NEXT CHARACTER
1192	637A	.	.	; DETERMINES WHICH GRAPHICS TABLE WILL BE USED
1193	637A	.	.	; ENTRY--DONT CARE
1194	637A	.	.	; EXIT---ALL REGISTERS DESTROYED
1195	637A	.	.	;*****
1196	637A	.	.	GSETUP EQU \$
1197	637A	AF	.	XRA A ;CLEAR PARAMETER FLAGS
1198	637B	32	B6 90	STA PRMDEX
1199	637E	3E	7F .	MVI A,-1-RESET ;CLEAR ALL PARAMETER FLAGS
1200	6380	CD	6D A2	CALL CLFLG7 ;EXCEPT THE RESET FLAG
1201	6383	21	76 60	LXI H,GTAB ;SET NEW RANGE TABLE
1202	6386	.	.	SETRTB EQU \$
1203	6386	22	D2 FF	SHLD ZRNGTA
1204	6389	21	D1 FF	LXI H,ZESCFG ;STOP 2-CHAR ESC SEQ COUNTER
1205	638C	36	FF .	MVI M,-1
1206	638E	C3	BD 00	JMP ZCRADV ;CLEAR CURSOR ADVANCE FLAG
1207	6391	.	.	;*****
1208	6391	.	.	; PLTSEQ--ESC * P RECEIVED, SET UP FOR PLOTTING
1209	6391	.	.	;*****
1210	6391	.	.	PLTSEQ EQU \$
1211	6391	21	A2 60	LXI H,PLTTAB ;SET NEW RANGE TABLE
1212	6394	22	D2 FF	SHLD ZRNGTA
1213	6397	21	55 64	LXI H,ASCABS ;DEFAULT TO ABSOLUTE ASCII
1214	639A	22	9C 90	SHLD PRMVEC ;POINTER TO PARAMETER ROUTIN
1215	639D	3E	40 .	MVI A,ASCII ;SET ASCII VECTOR FLAG
1216	639F	CD	67 A2	CALL STFLG7
1217	63A2	AF	.	XRA A
1218	63A3	32	D1 90	STA OCTANT
1219	63A6	3E	01 .	MVI A,SUPRO ;SUPRESS THE CURSOR
1220	63A8	CD	8A 9E	CALL SUPRGC
1221	63AB	C3	82 B8	JMP APLTOF ;TURN AUTO PLOT OFF
1222	63AE	.	.	;*****
1223	63AE	.	.	; PLTPRM--VECTOR PLOTTING PARAMETER RECEIVED
1224	63AE	.	.	; USE JUMP VECTOR TO PROCESS ACCORDING TO
1225	63AE	.	.	; PROPER FORMAT
1226	63AE	.	.	;*****
1227	63AE	.	.	PLTPRM EQU \$
1228	63AE	2A	9C 90	LHLD PRMVEC ;FETCH JUMP VECTOR
1229	63B1	E9	.	PCHL ;DO THE JUMP
1230	63B2	.	.	;*****
1231	63B2	.	.	; ASABFT--USE ASCII ABSOLUTE FORMAT
1232	63B2	.	.	;*****
1233	63B2	.	.	ASABFT EQU \$
1234	63B2	21	55 64	LXI H,ASCABS
1235	63B5	C3	C1 63	JMP ASC010
1236	63B8	.	.	;*****
1237	63B8	.	.	; ASINFT--USE ASCII INCREMENTAL FORMAT
1238	63B8	.	.	;*****
1239	63B8	.	.	ASINFT EQU \$

13255
2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

```
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE	36
1240	63B8	21	69	64	LXI H,ASCINC		
1241	63B8	C3	C1	63	JMP ASC010		

```
=====
```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
1243     63BE      . . .      ;*****
1244     63BE      . . .      ; ASRLFT--USE ASCII RELOCATABLE FORMAT
1245     63BE      . . .      ;*****
1246     63BE      . . .      ASRLFT EQU $
1247     63BE      21 87 64    LXI H,ASCREL
1248     63C1      . . .      ASC010 EQU $
1249     63C1      . . .      ; COMMAND RECEIVED--TERMINATE ANY POSSIBLE
1250     63C1      . . .      ; ASCII VECTOR IN PROGRESS
1251     63C1      E5 . . .    PUSH H          ;SAVE JUMP VECTOR
1252     63C2      CD A5 64    CALL ASCEND     ;FINISH ASCII VECTOR
1253     63C5      E1 . . .    POP H          ;RESTORE POINTER
1254     63C6      22 9C 90    SHLD PRMVEC     ;STORE VECTOR ROUTINE POINTE
1255     63C9      3E 40 .     MVI A,ASCII     ;SET ASCII VECTOR IN PROGRES
1256     63CB      CD 67 A2    CALL STFLG7
1257     63CE      C3 C1 99    JMP GEXIT
1258     63D1      . . .      ;*****
1259     63D1      . . .      ; ABSFMT--SWITCH TO ABSOLUTE FORMAT
1260     63D1      . . .      ;*****
1261     63D1      . . .      ABSFMT EQU $
1262     63D1      21 B7 64    LXI H,ABPARM
1263     63D4      C3 E6 63    JMP FMT010
1264     63D7      . . .      ;*****
1265     63D7      . . .      ; SHTFMT--SWITCH TO SHORT INCREMENTAL FORMAT
1266     63D7      . . .      ;*****
1267     63D7      . . .      SHTFMT EQU $
1268     63D7      21 DD 64    LXI H,SHTPRM
1269     63DA      C3 E6 63    JMP FMT010
1270     63DD      . . .      ;*****
1271     63DD      . . .      ; INCFMT--SWITCH TO INCREMENTAL FORMAT
1272     63DD      . . .      ;*****
1273     63DD      . . .      INCFMT EQU $
1274     63DD      21 09 65    LXI H,INCPRM
1275     63E0      C3 E6 63    JMP FMT010
1276     63E3      . . .      ;*****
1277     63E3      . . .      ; RELFMT--SWITCH TO RELOCATABLE FORMAT
1278     63E3      . . .      ;*****
1279     63E3      . . .      RELFMT EQU $
1280     63E3      21 2D 65    LXI H,RELPRM
1281     63E6      . . .      FMT010 EQU $
1282     63E6      E5 . . .    PUSH H          ;SAVE VECTOR ROUTINE PTR
1283     63E7      CD A5 64    CALL ASCEND     ;FINISH POSSIBLE ASCII VEC
1284     63EA      E1 . . .    POP H
1285     63EB      22 9C 90    SHLD PRMVEC     ;STORE POINTER TO ROUTINE
1286     63EE      3E 40 .     MVI A,ASCII     ;CLEAR ASCII VECTOR FLAG
1287     63F0      CD 6D A2    CALL CLFLG7
1288     63F3      C3 C1 99    JMP GEXIT
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 38
1290	63F6	.	.	. ;*****	
1291	63F6	.	.	. ; VECTOR COMMANDS COMMON TO ALL 3 ESCAPE	
1292	63F6	.	.	. ; SEQUENCES:	
1293	63F6	.	.	. ; PENDN--LOWER PEN	
1294	63F6	.	.	. ; PENUP--RAISE PEN--DO MOVE IN MIDDLE OF ESC SEQ	
1295	63F6	.	.	. ; USEGC--USE GRAPHICS CURSOR COORDINATE AS ENDPOINT	
1296	63F6	.	.	. ; ONEDOT--DRAW SINGLE DOT AT CURRENT POINT	
1297	63F6	.	.	. ;*****	
1298	63F6	.	.	. PENDN EQU \$	
1299	63F6	CD	A5 64	CALL ASCEND ;FINISH POSSIBLE ASCII VECTO	
1300	63F9	3E	01 .	MVI A,MOVE ;CLEAR MOVE FLAG	
1301	63FB	CD	2C A2	CALL CLFLG1	
1302	63FE	C3	C1 99	JMP GEXIT	
1303	6401	.	.	. PENUP EQU \$	
1304	6401	CD	A5 64	CALL ASCEND ;FINISH POSSIBLE ASCII VECTO	
1305	6404	3E	01 .	MVI A,MOVE ;SET MOVE FLAG	
1306	6406	CD	26 A2	CALL STFLG1	
1307	6409	C3	C1 99	JMP GEXIT	
1308	640C	.	.	. USEGC EQU \$	
1309	640C	CD	A5 64	CALL ASCEND ;FINISH POSSIBLE ASCII VECTO	
1310	640F	2A	CF 90	LHLD NEWGCX ;SET NEW POINT EQUAL TO	
1311	6412	22	DA 90	SHLD XNEW ;CURSOR POSITION	
1312	6415	2A	CD 90	LHLD NEWGCY	
1313	6418	22	D8 90	SHLD YNEW	
1314	641B	CD	DB 65	CALL VECTOR	
1315	641E	C3	C1 99	JMP GEXIT	
1316	6421	.	.	. ONEDOT EQU \$	
1317	6421	CD	A5 64	CALL ASCEND ;FINISH POSSIBLE ASCII VECTO	
1318	6424	CD	2A 64	CALL ONEDT1	
1319	6427	C3	C1 99	JMP GEXIT	
1320	642A	.	.	. ONEDT1 EQU \$;(INTERNAL ENTRY)	
1321	642A	3E	01 .	MVI A,MOVE ;CLEAR THE MOVE FLAG	
1322	642C	CD	2C A2	CALL CLFLG1	
1323	642F	3E	08 .	MVI A,DWFRST ;SET DRAW FIRST DOT FLAG	
1324	6431	CD	40 A2	CALL STFLG5	
1325	6434	2A	DE 90	LHLD XCURR ;SET NEW POINT = CURRENT	
1326	6437	22	DA 90	SHLD XNEW ;POINT	
1327	643A	2A	DC 90	LHLD YCURR	
1328	643D	22	D8 90	SHLD YNEW	
1329	6440	C3	E9 65	JMP VECTRO ;DRAW THE DOT	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 39
1331	6443	. . .	PNORG1 EQU \$	
1332	6443	CD A5 64	CALL ASCEND ;FINISH POSSIBLE ASCII VECTO	
1333	6446	C3 44 73	JMP PENORG ;SET RELOC ORG = CUR PT.	
1334	6449	. . .	NOP1 EQU \$	
1335	6449	CD A5 64	CALL ASCEND ;FINISH POSSIBLE ASCII VECTO	
1336	644C	C3 C1 99	JMP NOP ;PROCESS THE NOP	
1337	644F	. . .	*****	
1338	644F	. . .	; PLTESC--ESC RECEIVED IN PLOT SEQUENCE	
1339	644F	. . .	*****	
1340	644F	. . .	PLTESC EQU \$	
1341	644F	CD AB 9E	CALL ENABO ;RE-ENABLE THE CURSOR	
1342	6452	C3 B7 00	JMP ZESCAP ;PROCESS THE ESCAPE	
1343	6455	. . .	*****	
1344	6455	. . .	; ASCABS--PROCESS PARAMETER FOR ASCII ABSOLUTE	
1345	6455	. . .	; VECTOR. DRAW VECTOR WHEN TWO VALUES HAVE	
1346	6455	. . .	; BEEN RECEIVED	
1347	6455	. . .	*****	
1348	6455	. . .	ASCABS EQU \$	
1349	6455	CD AF 64	CALL GETVEC ;GET VECTOR PARAMETER	
1350	6458	F8 . .	RM ;NO	
1351	6459	C8 . .	RZ ;NO	
1352	645A	. . .	; HAVE NEW X AND Y NOW	
1353	645A	2A B9 90	LHLD PRMBUF ;LOAD X	
1354	645D	22 DA 90	SHLD XNEW	
1355	6460	2A BB 90	LHLD PRMBUF+2 ;LOAD Y	
1356	6463	22 D8 90	SHLD YNEW	
1357	6466	C3 DB 65	JMP VECTOR ;DRAW THE VECTOR	
1358	6469	. . .	*****	
1359	6469	. . .	; ASCINC--PROCESS ASCII INCREMENTAL VECTOR	
1360	6469	. . .	*****	
1361	6469	. . .	ASCINC EQU \$	
1362	6469	CD AF 64	CALL GETVEC ;GET VECTOR PARAMETER	
1363	646C	F8 . .	RM ;NO	
1364	646D	C8 . .	RZ ;NO	
1365	646E	. . .	; HAVE DELTA X AND DELTA Y	
1366	646E	2A B9 90	LHLD PRMBUF ;LOAD DELTA X	
1367	6471	EB . .	XCHG	
1368	6472	2A DE 90	LHLD XCURR ;ADD TO CURRENT X	
1369	6475	19 . .	DAD D ;TO GET NEW POINT	
1370	6476	22 DA 90	SHLD XNEW	
1371	6479	2A BB 90	LHLD PRMBUF+2 ;LOAD DELTA Y	
1372	647C	EB . .	XCHG	
1373	647D	2A DC 90	LHLD YCURR ;ADD TO CURRENT Y	
1374	6480	19 . .	DAD D	
1375	6481	22 D8 90	SHLD YNEW ;TO GET NEW Y	
1376	6484	C3 DB 65	JMP VECTOR ;DRAW THE VECTOR	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE	40
1378	6487	.	.	*****		
1379	6487	.	.	; ASCREL--PROCESS ASCII RELOCATABLE VECTOR		
1380	6487	.	.	*****		
1381	6487	.	.	ASCREL EQU \$		
1382	6487	CD	AF 64	CALL GETVEC ;GET VECTOR PARAMETER		
1383	648A	F8	.	RM ;NO		
1384	648B	C8	.	RZ ;NO		
1385	648C	.	.	; HAVE RELOCATABLE X AND Y		
1386	648C	2A	B9 90	LHLD PRMBUF ;LOAD X		
1387	648F	EB	.	XCHG		
1388	6490	2A	9A 90	LHLD XORG ;ADD RELOC ORG		
1389	6493	19	.	DAD D		
1390	6494	22	DA 90	SHLD XNEW		
1391	6497	2A	BB 90	LHLD PRMBUF+2		
1392	649A	EB	.	XCHG		
1393	649B	2A	98 90	LHLD YORG ;ADD RELOC ORG		
1394	649E	19	.	DAD D		
1395	649F	22	DB 90	SHLD YNEW		
1396	64A2	C3	DB 65	JMP VECTOR ;DRAW THE VECTOR		
1397	64A5	.	.	*****		
1398	64A5	.	.	; ASCEND--TERMINATE POSSIBLE ASCII VECTOR		
1399	64A5	.	.	; IN PROGRESS.		
1400	64A5	.	.	*****		
1401	64A5	.	.	ASCEND EQU \$		
1402	64A5	3A	96 90	LDA GFLGS7 ;ASCII VECTOR IN PROGRESS		
1403	64A8	E6	40 .	ANI ASCII		
1404	64AA	C8	.	RZ ;NO, DONE		
1405	64AB	2A	9C 90	LHLD PRMVEC ;YES, DRAW IT IF PARAMETER		
1406	64AE	E9	.	PCHL ;WAS BEING BUILT		
1407	64AF	.	.	*****		
1408	64AF	.	.	; GETVEC--GET VECTOR PARAMETER BYTE, AND TEST FOR		
1409	64AF	.	.	; 2 COMPLETE PARAMETERS RECEIVED		
1410	64AF	.	.	; EXIT P,NZ => HAVE BOTH BYTES		
1411	64AF	.	.	*****		
1412	64AF	.	.	GETVEC EQU \$		
1413	64AF	CD	A6 6C	CALL GPARAM ;PROCESS THE PARAMETER		
1414	64B2	3A	B6 90	LDA PRMDEX ;FETCH COMPLETED PARM. COUNT		
1415	64B5	3D	.	DCR A ;TEST FOR 2		
1416	64B6	C9	.	RET		

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 41
=====
1418     64B7     . . .      ;*****
1419     64B7     . . .      ;ABPARM--PARAMETER FOR ABSOLUTE VECTOR RECEIVED
1420     64B7     . . .      ; PARAMETERS NEEDED BY VECTOR ROUTINE ARE
1421     64B7     . . .      ; COMPUTED AS COORDINATE PARAMETERS ARRIVE
1422     64B7     . . .      ;*****
1423     64B7     . . .      ABPARM EQU $
1424     64B7     CD F8 A2    CALL PRMSTR      ;PROCESS PARAMETER
1425     64BA     21 B6 90    LXI H,PRMDEX    ;INCREMENT PARAMETER COUNT
1426     64BD     34 . .     INR M
1427     64BE     7E . .     MOV A,M          ;CHECK VALUE
1428     64BF     FE 04 .    CPI 4            ;HAVE ALL 4 YET?
1429     64C1     CA D1 64    JZ YABS          ;YES, PROCESS Y AND DRAW
1430     64C4     FE 02 .    CPI 2            ;HAVE COMPLETE X?
1431     64C6     C0 . .     RNZ              ;NO, WAIT FOR MORE PARAMETER
1432     64C7     . . .      ;HAVE COMPLETE X COORDINATE--REFORMAT IT AND
1433     64C7     . . .      ;COMPUTE DELTA X
1434     64C7     . . .      XABS EQU $
1435     64C7     21 B9 90    LXI H,PRMBUF    ;PTR TO X PARAMETERS
1436     64CA     CD D1 A2    CALL FORMAT      ;REFORMAT THEM
1437     64CD     22 DA 90    SHLD XNEW        ;STORE NEW X COORD
1438     64D0     C9 . .     RET
1439     64D1     . . .      ;HAVE COMPLETE Y COORD--REFORMAT IT, COMPUTE
1440     64D1     . . .      ;DELTA Y, AND DRAW THE VECTOR
1441     64D1     . . .      YABS EQU $
1442     64D1     21 BB 90    LXI H,PRMBUF+2  ;PTR TO Y PARAMETERS
1443     64D4     CD D1 A2    CALL FORMAT      ;REFORMAT THEM
1444     64D7     22 D8 90    SHLD YNEW        ;STORE NEW Y COORD
1445     64DA     C3 DB 65    JMP VECTOR       ;DRAW THE VECTOR
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1447	64DD	.	.	;*****	42
1448	64DD	.	.	; SHTPRM--PARAMETER FOR SHORT INCREMENTAL VECTOR	
1449	64DD	.	.	;RECEIVED	
1450	64DD	.	.	;*****	
1451	64DD	.	.	SHTPRM EQU \$	
1452	64DD	3A	88 FF	LDA ZCHAR ;FETCH PARAMETER	
1453	64E0	E6	1F .	ANI 370 ;WANT 5 LSB ONLY	
1454	64E2	.	.	;EXTEND SIGN BIT (BIT 4). WANT 16 BIT INCREMENT	
1455	64E2	26	00 .	MVI H,0 ;ASSUME +, SET MSBYTE TO 0	
1456	64E4	FE	10 .	CPI 200 ;IS SIGN BIT SET?	
1457	64E6	DA	EC 64	JC INC010 ;NO--OK AS IS	
1458	64E9	25	. .	DCR H ;YES, SET LEADING 1'S FOR -	
1459	64EA	F6	E0 .	ORI 3400 ;LEADING ONES IN A	
1460	64EC	.	.	INC010 EQU \$	
1461	64EC	6F	. .	MOV L,A ;HL=16 BIT DX OR DY INCREMEN	
1462	64ED	EB	. .	XCHG	
1463	64EE	21	B6 90	LXI H,PRMDEX ;FIND OUT WHICH ONE (X OR Y)	
1464	64F1	34	. .	INR M	
1465	64F2	7E	. .	MOV A,M ;FETCH INDEX	
1466	64F3	3D	. .	DCR A ;FIRST PARAMETER IS DELTA X	
1467	64F4	C2	FF 64	JNZ YINC ;SECOND IS DELTA Y	
1468	64F7	.	.	XINC EQU \$	
1469	64F7	.	.	;HAVE RECEIVED X INCREMENT (IN DE)	
1470	64F7	2A	DE 90	LHLD XCURR ;COMPUTE NEW POINT BY	
1471	64FA	19	. .	DAD D ;XNEW = XCURRENT + DELTAX	
1472	64FB	22	DA 90	SHLD XNEW	
1473	64FE	C9	. .	RET	
1474	64FF	.	.	YINC EQU \$	
1475	64FF	.	.	;NOW HAVE COMPLETE Y INCREMENT (IN DE)	
1476	64FF	2A	DC 90	LHLD YCURR ;COMPUTE YNEW BY	
1477	6502	19	. .	DAD D ;YNEW = YCURRENT + DELTAY	
1478	6503	22	D8 90	SHLD YNEW	
1479	6506	C3	DB 65	JMP VECTOR ;DRAW THE VECTOR	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 43
=====
1481      6509      . . .      ;*****
1482      6509      . . .      ; INCPRM--PARFAMETER FOR LONG INCREMENTAL VECTOR
1483      6509      . . .      ; RECEIVED. WHEN ALL 6 ARRIVE, DRAW THE VECTOR
1484      6509      . . .      ;*****
1485      6509      . . .      INCPRM EQU $
1486      6509      CD F8 A2    CALL PRMSTR      ;PROCESS THE PARAMETER
1487      650C      21 B6 90    LXI H,PRMDEX    ;SEE WHICH ONE IT IS
1488      650F      34 . .      INR M
1489      6510      7E . .      MOV A,M
1490      6511      FE 06 .      CPI 6           ;HAVE ALL 6 YET?
1491      6513      CA 23 65    JZ YLNG         ;YES, PROCESS Y AND DRAW
1492      6516      FE 03 .      CPI 3           ;HAVE COMPLETE X?
1493      6518      C0 . .      RNZ            ;NO, WAIT FOR MORE PARAMETER
1494      6519      . . .      ;HAVE COMPLETE X INCREMENT
1495      6519      . . .      XLNG EQU $
1496      6519      21 B9 90    LXI H,PRMBUF    ;PTR TO DX PARAMETERS
1497      651C      CD E3 A2    CALL LNGFMT     ;REFORMAT TO 16 BITS
1498      651F      EB . .      XCHG           ;DE = X INCREMENT
1499      6520      C3 F7 64    JMP XINC        ;REST SAME AS SHORT X INC
1500      6523      . . .      ;HAVE COMPLETE Y INCREMENT
1501      6523      . . .      YLNG EQU $
1502      6523      21 BC 90    LXI H,PRMBUF+3 ;PTR TO DY PARAMETERS
1503      6526      CD E3 A2    CALL LNGFMT     ;REFORMAT TO 16 BITS
1504      6529      EB . .      XCHG           ;DE = Y INCREMENT
1505      652A      C3 FF 64    JMP YINC        ;REST SAME AS SHORT Y INC
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  44
=====
1507      652D      .      .      .      ;*****
1508      652D      .      .      .      ; RELPRM--PARAMETER FOR RELOCATABLE VECTOR
1509      652D      .      .      .      ; RECEIVED.  WHEN 6 ARRIVE, DRAW THE VECTOR.
1510      652D      .      .      .      ;*****
1511      652D      .      .      .      RELPRM EQU $
1512      652D      CD  F8  A2      CALL PRMSTR      ;PROCESS THE PARAMETER
1513      6530      21  B6  90      LXI H,PRMDEX    ;UPDATE PARAMETER COUNT
1514      6533      34      .      .      INR M
1515      6534      7E      .      .      MOV A,M          ;FETCH COUNT
1516      6535      FE  06  .      CPI 6            ;HAVE ALL 6 ?
1517      6537      CA  4C  65      JZ RLP020       ;YES, PROCESS Y AND DRAW
1518      653A      FE  03  .      CPI 3            ;HAVE COMPLETE X COORD?
1519      653C      C0      .      .      RNZ              ;NO, WAIT FOR MORE PARAMETER
1520      653D      .      .      .      RLP010 EQU $
1521      653D      21  B9  90      LXI H,PRMBUF    ;PTR TO X PARAMETERS
1522      6540      CD  E3  A2      CALL LNGFMT     ;CONVERT TO 16 BIT VALUE
1523      6543      EB      .      .      XCHG
1524      6544      2A  9A  90      LHLD XORG       ;ADD VALUE OF RELOC ORIGIN
1525      6547      19      .      .      DAD D
1526      6548      22  DA  90      SHLD XNEW       ;STORE NEW X COORD
1527      6548      C9      .      .      RET
1528      654C      .      .      .      RLP020 EQU $
1529      654C      21  BC  90      LXI H,PRMBUF+3 ;PTR TO Y PARAMETERS
1530      654F      CD  E3  A2      CALL LNGFMT     ;CONVERT TO 16 BIT VALUE
1531      6552      EB      .      .      XCHG
1532      6553      2A  98  90      LHLD YORG       ;ADD VALUE OF RELOC Y ORG
1533      6556      19      .      .      DAD D
1534      6557      22  D8  90      SHLD YNEW       ;STORE NEW Y COORD
1535      655A      C3  DB  65      JMP VECTOR      ;DRAW THE VECTOR
=====

```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 45
1537	655D	. . .	*****	
1538	655D	. . .	; SETUP--COMPUTE PARAMETERS TO DRAW VECTOR	
1539	655D	. . .	; BETWEEN XCURR,YCURR AND XNEW,YNEW	
1540	655D	. . .	*****	
1541	655D	. . .	SETUP EQU \$	
1542	655D	AF . .	XRA A	
1543	655E	32 D1 90	STA OCTANT ;CLEAR VECTOR OCTANT	
1544	6561	2A DA 90	LHLD XNEW ;COMPUTE X BOUNDS CODE	
1545	6564	CD 9F 65	CALL XCODE	
1546	6567	F5 . .	PUSH PSW ;SAVE IT	
1547	6568	EB . .	XCHG	
1548	6569	2A DE 90	LHLD XCURR ;DE = NEW X, HL = CURRENT	
1549	656C	22 69 90	SHLD XSTART ;INITIALIZE STARTING POINT	
1550	656F	CD 8D 65	CALL DELXY ;COMPUTE DELTA X, OCTANT	
1551	6572	22 D6 90	SHLD DELTAX	
1552	6575	2A D8 90	LHLD YNEW ;COMPUTE Y BOUNDS CODE	
1553	6578	F1 . .	POP PSW ;RECALL X CODE	
1554	6579	CD 8D 65	CALL YCODE	
1555	657C	32 D3 90	STA NEWCD ;STORE BOUNDS CODE FOR X,Y	
1556	657F	EB . .	XCHG	
1557	6580	2A DC 90	LHLD YCURR ;DE = NEW Y, HL = CURRENT	
1558	6583	22 67 90	SHLD YSTART ;INITIALIZE STARTING POINT	
1559	6586	CD 8D 65	CALL DELXY ;COMPUTE DELTA Y, OCTANT	
1560	6589	22 D4 90	SHLD DELTAY	
1561	658C	C9 . .	RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 46
=====
1563     6580      . . .      ;*****
1564     6580      . . .      ;DELXY--COMPUTE DELTA = NEW-CURR, COMPUTE
1565     6580      . . .      ;SIGN BIT IN OCTANT WORD OF EITHER X OR Y COORD
1566     6580      . . .      ; FOR OCTANT, DELTA X MUST BE COMPUTED BEFORE
1567     6580      . . .      ; DELTA Y
1568     6580      . . .      ;ENTRY HL = CURRENT POINT,DE = NEWPOINT
1569     6580      . . .      ;EXIT HL = ABSOLUTE VALUE DELTA
1570     6580      . . .      ;*****
1571     6580      . . .      DELXY EQU $
1572     6580      . . .      ;COMPUTE DELTA
1573     6580      7B . .      MOV A,E          ;A=LSBYTE NEWPOINT
1574     658E      95 . .      SUB L            ;SUBTRACT LSBYTES
1575     658F      6F . .      MOV L,A
1576     6590      7A . .      MOV A,D          ;A=MSBYTE NEWPOINT
1577     6591      9C . .      SBB H            ;SUBTRACT MSBYTES
1578     6592      67 . .      MOV H,A          ;HL = NEWPT-CURRENTPT=DELTA
1579     6593      . . .      ;SIGN FLAG AND MSBIT SET IF DELTA IS NEGATIVE
1580     6593      07 . .      RLC              ;ROTATE SIGN OF DELTA INTO C
1581     6594      3A D1 90      LDA OCTANT       ;FETCH OCTANT WORD
1582     6597      17 . .      RAL              ;ROTATE SIGN OF INTO OCTANT
1583     6598      32 D1 90      STA OCTANT
1584     659B      . . .      ;COMPUTE ABSOLUTE VALUE OF DELTA
1585     659B      F0 . .      RP              ;DELTA IS ALREADY +
1586     659C      . . .      ; DELTA IS -, COMPLEMENT IT
1587     659C      C3 09 A3      JMP NEGATE
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
1589	659F	. . .	;*****	47
1590	659F	. . .	; XCODE--COMPUTE BOUNDS CODE FOR X COORD	
1591	659F	. . .	; (FROM NEWMAN & SPROULL, PAGE 123-124)	
1592	659F	. . .	; NOTE -- MAX AND MIN STORED AS -(MAX OR MIN)	
1593	659F	. . .	; ENTRY HL = X COORD	
1594	659F	. . .	; A = ANY PREVIOUS CODE	
1595	659F	. . .	; EXIT HL,DE,BC UNCHANGED	
1596	659F	. . .	; A = NEW CODE	
1597	659F	. . .	;*****	
1598	659F	. . .	XCODE EQU \$	
1599	659F	D5 . . .	PUSH D ;SAVE D	
1600	65A0	F5 . . .	PUSH PSW ;SAVE OLD CODE	
1601	65A1	EB . . .	XCHG ;DE = X COORD	
1602	65A2	2A 72 90	LHLD XMIN ;FETCH -XMIN	
1603	65A5	19 . . .	DAD D ;COMPUTE X - XMIN	
1604	65A6	. . .	; IF X - XMIN IS NEGATIVE, OUT OF BOUNDS	
1605	65A6	7C . . .	MOV A,H ;-?	
1606	65A7	87 . . .	ORA A	
1607	65A8	F2 B1 65	JP XCD010 ;IS +, TEST UPPER BOUND	
1608	65AB	. . .	; X IS .LT. MIN VALUE	
1609	65AB	F1 . . .	POP PSW ;FETCH OLD CODE	
1610	65AC	F6 01 .	ORI LTXMIN ;SET X-IS-LESS-THAN-MIN FLAG	
1611	65AE	EB . . .	XCHG ;RESTORE H	
1612	65AF	D1 . . .	POP D ;RECALL D	
1613	65B0	C9 . . .	RET	
1614	65B1	. . .	XCD010 EQU \$	
1615	65B1	. . .	; SEE IF X IS .GT. MAX	
1616	65B1	F1 . . .	POP PSW ;RESTORE OLD CODE	
1617	65B2	2A 70 90	LHLD XMAX ;FETCH -XMAX	
1618	65B5	2B . . .	DCX H ;WANT ONE BIGGER	
1619	65B6	19 . . .	DAD D ;COMPUTE X - (XMAX+1)	
1620	65B7	EB . . .	XCHG ;RESTORE HL	
1621	65B8	D1 . . .	POP D ;RESTORE D	
1622	65B9	. . .	; IF CARRY, X IS .GT. MAX	
1623	65B9	D0 . . .	RNC ;X IS OK	
1624	65BA	F6 02 .	ORI GTXMAX ;SET X-IS-GREATR-THAN-MAX FL	
1625	65BC	C9 . . .	RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 48
=====
1627      65BD      . . .      ;*****
1628      65BD      . . .      ; YCODE -- COMPUTE BOUNDS CODE FOR Y
1629      65BD      . . .      ; ENTRY, EXIT -- SAME AS X CODE
1630      65BD      . . .      ;*****
1631      65BD      . . .      YCODE EQU $
1632      65BD      D5 . . .      PUSH D ;SAVE D
1633      65BE      F5 . . .      PUSH PSW ;SAVE OLD CODE
1634      65BF      EB . . .      XCHG ;DE = Y COORD
1635      65C0      2A 6E 90    LHLD YMIN ;FETCH -YMIN
1636      65C3      19 . . .      DAD D ;COMPUTE Y-YMIN
1637      65C4      . . .      ; IF RESULT IS -, Y IS .LT. MIN
1638      65C4      7C . . .      MOV A,H ;TEST SIGN
1639      65C5      B7 . . .      ORA A
1640      65C6      F2 CF 65    JP YCD010 ;IS +, TEST UPPER BOUND
1641      65C9      . . .      ; Y IS .LT. MIN
1642      65C9      F1 . . .      POP PSW ;RESTORE OLD CODE
1643      65CA      F6 04 . . .      ORI LTYMIN ;SET Y-IS-LESS-THAN-MIN FLAG
1644      65CC      EB . . .      XCHG ;RESTORE HL
1645      65CD      D1 . . .      POP D ;RESTORE D
1646      65CE      C9 . . .      RET
1647      65CF      . . .      YCD010 EQU $
1648      65CF      . . .      ; SEE IF Y IS .GT. MAX
1649      65CF      F1 . . .      POP PSW ;RESTORE OLD CODE
1650      65D0      2A 6C 90    LHLD YMAX ;FETCH -YMAX
1651      65D3      2B . . .      DCX H ;WANT ONE GREATER
1652      65D4      19 . . .      DAD D ;COMPUTE Y - (YMAX+1)
1653      65D5      EB . . .      XCHG ;RESTORE HL
1654      65D6      D1 . . .      POP D ;RESTORE D
1655      65D7      . . .      ; IF CARRY, Y IS .GT. MAX
1656      65D7      D0 . . .      RNC ;Y IS OK
1657      65D8      F6 08 . . .      ORI GTYMAX ;SET Y-IS-GREATR-THAN-MAX FL
1658      65DA      C9 . . .      RET
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
1660      65DB      .      .      .      ;*****
1661      65DB      .      .      .      ;
1662      65DB      .      .      .      ; VECTOR--DOES A DRAW OR MOVE BETWEEN CURRENT
1663      65DB      .      .      .      ; POINT AND NEW POINT.
1664      65DB      .      .      .      ;
1665      65DB      .      .      .      ;*****
1666      65DB      .      .      .      VECTOR EQU $
1667      65DB      3A  DB  FA      LDA  LNTYPE      ;IS POINT PLOT SELECTED?
1668      65DE      FE  08      CPI  PNTPLT
1669      65E0      C2  E9  65      JNZ  VECTRO      ;NO
1670      65E3      CD  2D  98      CALL CPUPD1      ;YES, MOVE TO NEW POINT
1671      65E6      C3  2A  64      JMP  ONEDT1      ;AND DRAW THE DOT
1672      65E9      .      .      .      VECTRO EQU $
1673      65E9      CD  7E  70      CALL RBOFF      ;TURN RB LINE OFF
1674      65EC      .      .      .      VECTR1 EQU $
1675      65EC      CD  5D  65      CALL SETUP      ;COMPUTE VECTOR PARAMETERS
1676      65EF      .      .      .      ;TEST FOR DRAW OR MOVE
1677      65EF      21  B2  90      LXI  H,GFLGS1
1678      65F2      3E  01      MVI  A,MOVE      ;MOVE BIT SET?
1679      65F4      A6  .      .      ANA  M
1680      65F5      CA  05  66      JZ   VEC010      ;NO, DO A DRAW
1681      65F8      .      .      .      ;DO A MOVE
1682      65F8      2F  .      .      CMA                      ;CLEAR THE MOVE FLAG
1683      65F9      A6  .      .      ANA  M
1684      65FA      F6  08      ORI  NEWWA      ;SET USE NEW WA FLAG
1685      65FC      77  .      .      MOV  M,A        ;STORE UPDATED FLAGS
1686      65FD      3E  08      MVI  A,DWFRST   ;SET THE DRAW FIRT DOT FLAG
1687      65FF      CD  40  A2      CALL STFLG5     ;AFTER A MOVE
1688      6602      C3  23  66      JMP  VEC030
1689      6605      .      .      .      VEC010 EQU $
1690      6605      .      .      .      ;DRAW VECTOR. FIRST, CHECK TO SEE IF EITHER
1691      6605      .      .      .      ;COMPLETELY ON OR COMPLETELY OFF SCREEN
1692      6605      2A  D2  90      LHL  CURCD      ;L=CUR. CODE, H = NEW CODE
1693      6608      .      .      .      ;IF BOUNDS CODES FOR BOTH NEWPOINT AND CURRENT
1694      6608      .      .      .      ;POINT ARE 0, VECTOR IS ENTIRELY VISIBLE
1695      6608      7C  .      .      MOV  A,H
1696      6609      B5  .      .      ORA  L          ;BOTH CODES ZERO?
1697      660A      CA  18  66      JZ   VEC020     ;YES--JUST DRAW VECTOR
1698      660D      .      .      .      ;IF LOGICAL AND OF CODES IS NOT ZERO, VECTOR
1699      660D      .      .      .      ;IS ENTIRELY OFF SCREEN
1700      660D      7C  .      .      MOV  A,H
1701      660E      A5  .      .      ANA  L          ;LOGICAL PRODUCT NOT ZERO?
1702      660F      C2  23  66      JNZ  VEC030     ;YES--JUST UPDATE ADDRESSES
1703      6612      .      .      .      ;VECTOR IS PARTIALLY OFF SCREEN, AND MUST BE
1704      6612      .      .      .      ;CLIPPED.DELTAX,DELTAY, XSTART,YSTART, AND OCTANT
1705      6612      .      .      .      ;ARE RE-COMPUTED.
1706      6612      CD  8E  67      CALL CLIP
1707      6615      .      .      .      ; JUST UPDATE ADDRESS IF VECTOR WAS COMPLETELY
1708      6615      .      .      .      ; OFFSCREEN
1709      6615      C2  23  66      JNZ  VEC030     ;JMP IF VECTOR INVISIBLE
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
1710      6618      .      .      .      VEC020 EQU $
1711      6618      3A      B2      90      LDA      GFLGS1      ;IS AREA PATTERN ON?
1712      6618      E6      04      .      ANI      AREAPT      ;IF SO, MUST FETCH PROPER
1713      661D      C4      AB      A3      CNZ      GETPAT      ;PATTERN BYTE
1714      6620      .      .      .      ;CALL THE ROUTINE TO SEND THE PARAMETERS TO THE
1715      6620      .      .      .      ;GRAPHICS CONTROLLER
1716      6620      CD      60      66      CALL     DRWVEC      ;DRAW THE VECTOR
1717      6623      .      .      .      VEC030 EQU $
1718      6623      .      .      .      ;UPDATE ADDRESSES AND VARIABLES
1719      6623      AF      .      .      XRA      A
1720      6624      32      B6      90      STA      PRMDEX      ;RESET PARAMETER INDEX
1721      6627      2A      DA      90      LHLD     XNEW        ;SET CURRENT POINT <= NEW
1722      662A      22      DE      90      SHLD     XCURR       ;POINT
1723      662D      2A      D8      90      LHLD     YNEW
1724      6630      22      DC      90      SHLD     YCURR
1725      6633      3A      D3      90      LDA      NEWCD        ;ALSO UPDATE BNDS CODE
1726      6636      32      D2      90      STA      CURCD
1727      6639      .      .      .      ; IF VECTOR WAS CLIPPED, SET NEW WA FLAG
1728      6639      21      B2      90      LXI      H,GFLGS1
1729      663C      7E      .      .      MOV      A,M          ;WAS IT CLIPPED?
1730      663D      E6      80      .      ANI      CLIPED
1731      663F      CA      47      66      JZ       VEC040      ;NO
1732      6642      2F      .      .      CMA
1733      6643      A6      .      .      ANA      M          ;YES, CLEAR CLIPPED FLAG
1734      6644      F6      08      .      ORI      NEWWA      ;SET NEW WA FLAG
1735      6646      77      .      .      MOV      M,A
1736      6647      .      .      .      VEC040 EQU $
1737      6647      .      .      .      ; UPDATE START OF LINE COORDINATES
1738      6647      3A      97      90      LDA      GFLGS6      ;SUPRESS SOL UPDATE?
1739      664A      E6      04      .      ANI      NOSOL
1740      664C      C2      60      A2      JNZ      CLFLG6      ;YES, CLEAR SUPRESS FLAG
1741      664F      2A      DE      90      LHLD     XCURR       ;SET SOL = CURRENT POINT
1742      6652      22      7B      90      SHLD     XSOL
1743      6655      2A      DC      90      LHLD     YCURR
1744      6658      22      79      90      SHLD     YSOL
1745      665B      3E      40      .      MVI      A,GCM4      ;OVERRIDE CURSOR LOC FOR CHA
1746      665D      C3      46      A2      JMP      CLFLG5
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 51
1748	6660	. . .	;*****	
1749	6660	. . .	;DRWVEC--SEND PARAMETERS TO GRAPHICS CONTROLLER	
1750	6660	. . .	;AND DRAW VECTOR. ASSUMES DELTAX,DELTAY,OCTANT,	
1751	6660	. . .	;ARE COMPUTED, STATING ADDRESS IS IN XSTART,	
1752	6660	. . .	;YSTART.	
1753	6660	. . .	;*****	
1754	6660	. . .	DRWVEC EQU \$	
1755	6660	. . .	;FIRST, SEE IF OLD WA (IN CONTROLLER) CAN BE USED	
1756	6660	3A B2 90	LDA GFLGS1 ;MUST NEW WRITE ADDRESS BE	
1757	6663	E6 08 .	ANI NEWWA ;COMPUTED?	
1758	6665	CA 78 66	JZ DRW010 ;NO	
1759	6668	. . .	;COMPUTE NEW WA FROM XSTART,YSTART.	
1760	6668	. . .	;COMPUTE NEW Y = (359-YSTART) * 45, SHIFT LEFT 4	
1761	6668	. . .	;PLACES, AND ADD XSTART	
1762	6668	2A 67 90	LHLD YSTART	
1763	666B	CD 5B 67	CALL MPY45 ;CONVERT Y	
1764	666E	EB . .	XCHG ;DE = CONVERTED Y	
1765	666F	2A 69 90	LHLD XSTART	
1766	6672	EB . .	XCHG ;HL = Y, DE = X	
1767	6673	CD 6F 67	CALL GETWA ;SHIFT Y AND ADD X	
1768	6676	F5 . .	PUSH PSW ;SAVE 6 MSBITS	
1769	6677	E5 . .	PUSH H ;SAVE 12 LSBITS	
1770	6678	. . .	DRW010 EQU \$	
1771	6678	. . .	;COMPUTE DELTAX-DELTAY, WANT SIGN ONLY	
1772	6678	2A D4 90	LHLD DELTAY ;HL = DELTA Y	
1773	667B	3A D6 90	LDA DELTAX	
1774	667E	5F . .	MOV E,A ;SET DE = DELTAX	
1775	667F	95 . .	SUB L ;SUBTRACT LSBYTES	
1776	6680	3A D7 90	LDA DELTAX+1	
1777	6683	57 . .	MOV D,A ;DE NOW = DELTA X	
1778	6684	9C . .	SBB H ;SUBTRACT MSBYTES	
1779	6685	. . .	;ADD SIGN OF DX-DY TO OCTANT WORD. SIGNS OF	
1780	6685	. . .	;DX,DY, AND DX-DY DETERMINES OCTANT OF VECTOR	
1781	6685	07 . .	RLC ;ROTATE MSBIT (SIGN) TO CY	
1782	6686	3A D1 90	LDA OCTANT	
1783	6689	17 . .	RAL ;ROTATE SIGN INTO OCTANT WOR	
1784	668A	47 . .	MOV B,A ;SAVE OCTANT IN B	
1785	668B	. . .	;IF SIGN OF DX-DY IS +,LEAVE DE = DELTAX,HL=DELTAY	
1786	668B	F2 8F 66	JP DRW015	
1787	668E	. . .	;ABSOLUTE SLOPE OF VECTOR > 45 DEGREES, MUST	
1788	668E	. . .	;INTERCHANGE DELTAX AND DELTAY IN CALCULATIONS	
1789	668E	EB . .	XCHG	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 52
=====
1791     668F      . . .      ;*****
1792     668F      . . .      ;CONTROLLER CAN NOW BE LOADED WITH VECTOR PARAMS.
1793     668F      . . .      ;MUST WAIT UNTIL IT IS FINISHED WITH WHAT IT IS
1794     668F      . . .      ;CURRENTLY DOING
1795     668F      . . .      ;*****
1796     668F      . . .      DRW015 EQU $
1797     668F      . . .      ; THE CURSOR MUST BE SUPRESSED
1798     668F      E5 . .      PUSH H          ;SAVE H AND D
1799     6690      D5 . .      PUSH D
1800     6691      3E 08 .      MVI A,TIMSUP   ;SET SUPPRESS BIT
1801     6693      CD 8A 9E     CALL SUPRGC     ;SUPPRESS THE GRAPHICS CURSOR
1802     6696      D1 . .      POP D          ;HW IS GUARENTEED TO BE IDLE
1803     6697      E1 . .      POP H          ;AFTER RETURN FROM SUPRGC
1804     6698      . . .      ;COMPUTE AND LOAD PARAMETERS
1805     6698      . . .      ;D1
1806     6698      29 . .      DAD H
1807     6699      22 1E 89     SHLD D1
1808     669C      . . .      ;COMPUTE -DE
1809     669C      AF . .      XRA A          ;SUBTRACT FROM 0 TO NEGATE
1810     669D      93 . .      SUB E          ;LSBYTE DONE
1811     669E      5F . .      MOV E,A
1812     669F      3E 00 .      MVI A,0
1813     66A1      9A . .      SBB D          ;MSBYTE DONE
1814     66A2      57 . .      MOV D,A       ;DE NOW = - DE
1815     66A3      . . .      ;INITIAL D
1816     66A3      19 . .      DAD D
1817     66A4      22 10 89     SHLD INITD
1818     66A7      . . .      ;D2
1819     66A7      19 . .      DAD D
1820     66A8      22 1C 89     SHLD D2
1821     66AB      . . .      ;DOT COUNT
1822     66AB      EB . .      XCHG
1823     66AC      2B . .      DCX H
1824     66AD      22 12 89     SHLD DC
1825     66B0      . . .      ;USING OCTANT, FETCH PROPER M1 AND M2 VALUES
1826     66B0      . . .      ;FROM TABLE. FIRST, MULTIPLY OCTANT BY 6
1827     66B0      78 . .      MOV A,B       ;FETCH OCTANT
1828     66B1      07 . .      RLC
1829     66B2      07 . .      RLC           ;A = A * 4
1830     66B3      80 . .      ADD B         ;A * 5
1831     66B4      80 . .      ADD B         ;A * 6
1832     66B5      4F . .      MOV C,A
1833     66B6      06 00 .      MVI B,0      ;BC = INDEX TO TABLE
1834     66B8      21 2B 67     LXI H,OCTTAB ;BASE OF TABLE
1835     66B8      09 . .      DAD B        ;HL = BASE + INDEX
1836     66BC      . . .      ;HL = POINTER TO FIRST PARAMETER FOR THIS OCTANT
1837     66BC      . . .      ;MUST LOAD M1, M2, AND SIGNS
1838     66BC      5E . .      MOV E,M
1839     66BD      23 . .      INX H
1840     66BE      56 . .      MOV D,M      ;DE = M1

```



```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 53
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
1841	66BF	EB	. .	XCHG
1842	66C0	22	1A 89	SHLD M1 ;SEND M1 TO HW
1843	66C3	EB	. .	XCHG
1844	66C4	23	. .	INX H
1845	66C5	5E	. .	MOV E,M
1846	66C6	23	. .	INX H
1847	66C7	56	. .	MOV D,M ;DE = M2
1848	66C8	EB	. .	XCHG
1849	66C9	22	16 89	SHLD M2 ;SEND M2 TO HW
1850	66CC	EB	. .	XCHG
1851	66CD	23	. .	INX H
1852	66CE	7E	. .	MOV A,M ;LOAD SIGN OF M1
1853	66CF	32	18 89	STA SIGNM1 ;SEND TO HW
1854	66D2	23	. .	INX H
1855	66D3	7E	. .	MOV A,M ;LOAD SIGN OF M2
1856	66D4	32	14 89	STA SIGNM2 ;SEND TO HW

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1858	66D7	.	.	;*****	54
1859	66D7	.	.	;NOW SEE IF A NEW WA WAS COMPUTED AND MUST BE	
1860	66D7	.	.	;SENT	
1861	66D7	.	.	;*****	
1862	66D7	21	B2 90	LXI H,GFLGS1 ;WAS USE NEW WA FLAG ON?	
1863	66DA	7E	.	MOV A,M	
1864	66DB	4F	.	MOV C,A ;SAVE GFLGS1	
1865	66DC	E6	08 .	ANI NEWWA ;TEST NEW WA FLAG	
1866	66DE	32	0B 89	STA SELWA ;SEND FLAG TO HW IF ON	
1867	66E1	CA	EF 66	JZ DRW020 ;JUMP IF NOT ON	
1868	66E4	.	.	;MUST SEND NEW WA TO CONTROLLER	
1869	66E4	2F	.	CMA ;CLEAR NEWWA FLAG	
1870	66E5	A6	.	ANA M	
1871	66E6	77	.	MOV M,A	
1872	66E7	E1	.	POP H ;RECALL 12 LSBITS	
1873	66E8	22	0E 89	SHLD LSBWA	
1874	66EB	F1	.	POP PSW ;RECALL 6 MSBITS	
1875	66EC	32	0C 89	STA MSBWA	
1876	66EF	.	.	DRW020 EQU \$	
1877	66EF	.	.	;SEND PATTERN,MODE, AND PRESCALE	
1878	66EF	3A	B5 90	LDA CURMOD ;SEND DRAWING MODE	
1879	66F2	32	41 89	STA HCEJK	
1880	66F5	.	.	;IF DRAW FIRST DOT FLAG IS ON, OR AREA PATTERN IS	
1881	66F5	.	.	;ON, SEND THE CURRENT PATTERN BYTE	
1882	66F5	3E	04 .	MVI A,AREAPT ;TEST AREA PATERN	
1883	66F7	A1	.	ANA C ;BIT IN GFLGS1	
1884	66F8	4F	.	MOV C,A	
1885	66F9	3A	AE 90	LDA GFLGSS ;TEST DRAW FIRST DOT FLAG	
1886	66FC	E6	08 .	ANI DWFRST	
1887	66FE	B1	.	ORA C ;IS EITHER ON??	
1888	66FF	CA	0E 67	JZ DRW030 ;NO--DONT CHANGE PATTERN	
1889	6702	3A	B4 90	LDA CURPAT ;YES--SEND NEW PATTERN BYTE	
1890	6705	32	40 89	STA PATERN	
1891	6708	3A	B3 90	LDA SCALE ;SET PRESCALE VALUE	
1892	670B	32	21 89	STA SCALER	
1893	670E	.	.	DRW030 EQU \$	
1894	670E	.	.	;SEND THE SELF TEST FLAGS. IF SELF TEST IS NOT IN	
1895	670E	.	.	;PROGRESS, THEY MUST BE 0 TO CLEAR THE ZOOM PARAM	
1896	670E	.	.	;ETERS WHICH USE THE SAME BUFFER LOCATIONS	
1897	670E	3A	A6 90	LDA STFLAG ;SEND SELF TEST ON/OFF FLAG	
1898	6711	32	09 89	STA SLFTST	
1899	6714	AF	.	XRA A ;DISABLE CONTINUE SELFTST	
1900	6715	32	07 89	STA CONTST	
1901	6718	.	.	;IF THE PREVIOUS COMMAND WAS A MOVE, THE DRAW	
1902	6718	.	.	;FIRST DOT FLAG IS SET, OTHERWISE, THE FIRST	
1903	6718	.	.	;DOT IS NOT DRAWN	
1904	6718	21	AE 90	LXI H,GFLGSS	
1905	671B	7E	.	MOV A,M ;FETCH DRAW FIRST DOT FLAG	
1906	671C	32	01 89	STA DRWDOT ;SEND TO HW	
1907	671F	E6	F7 .	ANI 377Q-DWFRST ;CLEAR THE FLAG AFTER	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
1908     6721     77 . .      MOV M,A      ;A DRAW
1909     6722     . . .      HWGO EQU $
1910     6722     3A B1 90    LDA GFLGS2   ;FETCH HW FLAGS
1911     6725     F6 01 .     ORI BUSY     ;SET THE DRAW BIT
1912     6727     32 20 89    STA HWFLGS   ;SEND TO HW
1913     672A     C9 . .      RET
1914     672B     . . .      ;*****
1915     672B     . . .      ;VALUES OF M1, M2, AND SIGNS KEYED BY OCTANT
1916     672B     . . .      ;STORED AS LSBYTE M1, MSBYTE M1, LSBYTE M2,
1917     672B     . . .      ;MSBYTE M2, SIGN M1, SIGN M2
1918     672B     . . .      ;*****
1919     672B     . . .      OCTTAB EQU $
1920     672B     . . .      ;OCT 1 M1=1, M2=-719
1921     672B     01 00 31    DB 001Q,000Q,061Q,375Q,000Q,377Q
1922     6731     . . .      ;OCT 2 M1=-720,M2=-719
1923     6731     30 FD 31    DB 060Q,375Q,061Q,375Q,377Q,377Q
1924     6737     . . .      ;OCT 8 M1=1, M2=721
1925     6737     01 00 D1    DB 001Q,000Q,321Q,002Q,000Q,000Q
1926     673D     . . .      ;OCT 7 M1=720, M2=721
1927     673D     D0 02 D1    DB 320Q,002Q,321Q,002Q,000Q,000Q
1928     6743     . . .      ;OCT 4 M1=-1, M2=-721
1929     6743     FF FF 2F    DB 377Q,377Q,057Q,375Q,377Q,377Q
1930     6749     . . .      ;OCT 3 M1=-720,M2=-721
1931     6749     30 FD 2F    DB 060Q,375Q,057Q,375Q,377Q,377Q
1932     674F     . . .      ;OCT 5 M1=-1, M2=719
1933     674F     FF FF CF    DB 377Q,377Q,317Q,002Q,377Q,000Q
1934     6755     . . .      ;OCT 6 M1=720, M2=719
1935     6755     D0 02 CF    DB 320Q,002Q,317Q,002Q,000Q,000Q
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 56
=====
1937      675B      . . .      ;*****
1938      675B      . . .      ;MPY45--HL = (359-HL) * 45
1939      675B      . . .      ;THIS IS TO CONVERT Y COORD TO GRAPHICS ADDRESS
1940      675B      . . .      ; A, DE DESTROYED
1941      675B      . . .      ;*****
1942      675B      . . .      MPY45 EQU $
1943      675B      . . .      ;COMPUTE 359 - HL
1944      675B      3E 67 .      MVI A,147Q      ;LSBYTE OF 359
1945      675D      95 . . .      SUB L
1946      675E      6F . . .      MOV L,A
1947      675F      5F . . .      MOV E,A
1948      6760      3E 01 .      MVI A,1Q      ;MSBYTE OF 359
1949      6762      9C . . .      SBB H
1950      6763      67 . . .      MOV H,A
1951      6764      57 . . .      MOV D,A      ;DE,HL = 359 - HL
1952      6765      . . .      ; COMPUTE 45 * HL
1953      6765      29 . . .      DAD H      ;2*HL
1954      6766      29 . . .      DAD H      ;4*HL
1955      6767      19 . . .      DAD D      ;5*HL
1956      6768      E5 . . .      PUSH H      ;SAVE 5*HL
1957      6769      29 . . .      DAD H      ;10*HL
1958      676A      29 . . .      DAD H      ;20*HL
1959      676B      29 . . .      DAD H      ;40*HL
1960      676C      D1 . . .      POP D      ;DE=5*HL
1961      676D      19 . . .      DAD D      ;HL=(359-HL)*45
1962      676E      C9 . . .      RET
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 57
1964	676F	. . .	;*****	
1965	676F	. . .	;GETWA--AN 18 BIT WRITE ADDRESS IS COMPUTED FROM	
1966	676F	. . .	;X,Y COORDINATES	
1967	676F	. . .	;ENTRY HL = (359-Y) * 45	
1968	676F	. . .	; DE = X	
1969	676F	. . .	;EXIT HL = 12 LSBITS	
1970	676F	. . .	; A = 6 MSBITS	
1971	676F	. . .	; DE DESTROYED	
1972	676F	. . .	;*****	
1973	676F	. . .	GETWA EQU \$	
1974	676F	C5 . . .	PUSH B ;SAVE B	
1975	6770	. . .	; B IS USED FOR EXTENSION OF SIGN OF X	
1976	6770	06 00 .	MVI B,0 ;ASSUME X IS +	
1977	6772	7A . . .	MOV A,D ;CHECK SIGN	
1978	6773	B7 . . .	ORA A ;IS IT REALLY?	
1979	6774	F2 78 67	JP GWA010 ;YES	
1980	6777	05 . . .	DCR B ;NO, SET B = -1	
1981	6778	. . .	GWA010 EQU \$	
1982	6778	. . .	;COMPUTE WA BY SHIFTING Y LEFT 4 PLACES AND ADDING	
1983	6778	. . .	;X	
1984	6778	AF . . .	XRA A ;A = DESTINATION OF MSBITS	
1985	6779	. . .	;SHIFT HL LEFT 4 PLACES, SAVING OVERFLO IN A	
1986	6779	29 . . .	DAD H	
1987	677A	17 . . .	RAL	
1988	677B	29 . . .	DAD H	
1989	677C	17 . . .	RAL	
1990	677D	29 . . .	DAD H	
1991	677E	17 . . .	RAL	
1992	677F	29 . . .	DAD H	
1993	6780	17 . . .	RAL	
1994	6781	. . .	;NOW ADD X	
1995	6781	19 . . .	DAD D	
1996	6782	88 . . .	ADC B ;ADD MSBYTE OF X + CY	
1997	6783	. . .	;NOW, MUST MERGE BITS 12-15 WITH BITS 16,17	
1998	6783	5F . . .	MOV E,A ;SAVE A (BITS 16,17)	
1999	6784	7C . . .	MOV A,H ;A = BITS 8-15	
2000	6785	E6 F0 .	ANI 360Q ;HAVE BITS 12-15	
2001	6787	B3 . . .	ORA E ;MERGE WITH 16,17	
2002	6788	. . .	;A=15-14-13-12-0-0-17-16	
2003	6788	0F . . .	RRC ;ROTATE 4 TIMES TO	
2004	6789	0F . . .	RRC ;PUT IN PROPER	
2005	678A	0F . . .	RRC ;ORDER	
2006	678B	0F . . .	RRC	
2007	678C	. . .	;A = 12-17, HL = 0-15 (OVERLAP DOESNT MATTER)	
2008	678C	C1 . . .	POP B ;RESTORE B	
2009	678D	C9 . . .	RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 58
=====
2011      678E      .      .      .      ;*****
2012      678E      .      .      .      ; CLIP--ONE OR BOTH ENDPOINTS IS OFF SCREEN
2013      678E      .      .      .      ; ADJUST XSTART,YSTART, DELTAX&Y,OCTANT FOR
2014      678E      .      .      .      ; A NEW VECTOR WHICH IS ENTIRELY ON SCREEN
2015      678E      .      .      .      ; METHOD USED IS DESCRIBED IN "A CLIPPING
2016      678E      .      .      .      ; DIVIDER", AFIPS 1968 FALL JOINT COMPUTER
2017      678E      .      .      .      ; CONFERENCE, VOL 33, PART 1, PAGES 765-775
2018      678E      .      .      .      ; EXIT  NZ => VECTOR IS COMPLETELY OFF SCREEN
2019      678E      .      .      .      ;          CLIPED FLAG IN GFLGS1 IS SET
2020      678E      .      .      .      ;          NEW POINT, CURRENT POINT, NEWCD, CURCD
2021      678E      .      .      .      ;          ARE UNCHANGED
2022      678E      .      .      .      ;*****
2023      678E      .      .      .      CLIP  EQU  $
2024      678E      21  B2  90      LXI  H,GFLGS1  ;SET CLIPPED AND NEW
2025      6791      3E  88  .      MVI  A,CLIPED+NEWWA  ;WA FLAGS
2026      6793      B6  .      .      ORA  M
2027      6794      77  .      .      MOV  M,A
2028      6795      .      .      .      ; ASSUME CURRENT START AND ENDING POINTS WILL
2029      6795      .      .      .      ; BE UNCHANGED, AND COMPUTE DELTA X,Y
2030      6795      2A  DE  90      LHLD XCURR
2031      6798      22  69  90      SHLD XSTART      ;NEW STARTING X
2032      679B      EB  .      .      XCHG              ;DE = X CURR
2033      679C      2A  DA  90      LHLD XNEW
2034      679F      22  65  90      SHLD XFIN        ;NEW ENDING X
2035      67A2      CD  09  A3      CALL NEGATE      ;HL = -XNEW
2036      67A5      .      .      .      ; COMPUTE DELTA X
2037      67A5      19  .      .      DAD  D            ;HL = XCURR-XNEW
2038      67A6      22  5D  90      SHLD XDEL        ;STORE DELTA X
2039      67A9      2A  DC  90      LHLD YCURR
2040      67AC      22  67  90      SHLD YSTART      ;NEW STARTING Y
2041      67AF      EB  .      .      XCHG              ;DE = YCURR
2042      67B0      2A  D8  90      LHLD YNEW
2043      67B3      22  63  90      SHLD YFIN        ;NEW ENDING Y
2044      67B6      CD  09  A3      CALL NEGATE      ;HL = -YNEW
2045      67B9      .      .      .      ; COMPUTE DELTA Y
2046      67B9      19  .      .      DAD  D            ;HL = YCURR-YNEW
2047      67BA      22  5B  90      SHLD YDEL        ;STORE DELTA Y
2048      67BD      .      .      .      ; SEE IF NEW POINT IS IN BOUNDS
2049      67BD      3A  D3  90      LDA  NEWCD        ;GET BOUNDS CODE
2050      67C0      B7  .      .      ORA  A            ;ON SCREEN?
2051      67C1      CA  D6  67      JZ   CLP010      ;YES, LEAVE FIN PT. AS IS
2052      67C4      .      .      .      ; NEW POINT OUT, COMPUTE NEW FINISH POINT
2053      67C4      47  .      .      MOV  B,A          ;PUT BOUNDS CODE IN B
2054      67C5      2A  D8  90      LHLD YNEW
2055      67C8      EB  .      .      XCHG
2056      67C9      2A  DA  90      LHLD XNEW        ;HL = X, DE = Y
2057      67CC      CD  3A  68      CALL CLPALG      ;DO THE CLIPPING
2058      67CF      22  65  90      SHLD XFIN        ;STORE NEW ENDING POINT
2059      67D2      EB  .      .      XCHG
2060      67D3      22  63  90      SHLD YFIN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2061	67D6	.	.	CLP010 EQU \$	59
2062	67D6	.	.	; SEE IF CURRENT POINT IS IN BOUNDS	
2063	67D6	3A	D2 90	LDA CURCD ;GET BOUNDS CODE	
2064	67D9	B7	.	ORA A ;ON SCREEN?	
2065	67DA	CA	09 68	JZ CLP020 ;YES, LEAVE START PT. AS IS	
2066	67DD	.	.	; CURRENT POINT OUT, COMPUTE NEW START POINT	
2067	67DD	.	.	; FIRST, REVERSE SENSE OF DELTA X,Y	
2068	67DD	47	.	MOV B,A ;PUT BOUNDS CODE IN B	
2069	67DE	2A	5D 90	LHLD XDEL	
2070	67E1	CD	09 A3	CALL NEGATE ;NEGATE DELTA X	
2071	67E4	22	5D 90	SHLD XDEL	
2072	67E7	2A	5B 90	LHLD YDEL	
2073	67EA	CD	09 A3	CALL NEGATE ;NEGATE DELTA Y	
2074	67ED	22	5B 90	SHLD YDEL	
2075	67F0	2A	DC 90	LHLD YCURR	
2076	67F3	EB	.	XCHG	
2077	67F4	2A	DE 90	LHLD XCURR ;HL = X, DE = Y	
2078	67F7	CD	3A 68	CALL CLPALG ;DO THE CLIPPING	
2079	67FA	.	.	*****	
2080	67FA	.	.	; ROM BREAK 1	
2081	67FA	C3	02 68	JMP ZBRK1C	
2082	67FD	.	.	ORG BEGIN+40000	
2083	6800	.	.	ZBRK1 EQU \$	
2084	6800	54	.	DB VERSN	
2085	6801	68	.	DB ZBRK1/256	
2086	6802	.	.	ZBRK1C EQU \$	
2087	6802	.	.	*****	
2088	6802	22	69 90	SHLD XSTART ;STORE NEW START POINT	
2089	6805	EB	.	XCHG	
2090	6806	22	67 90	SHLD YSTART	
2091	6809	.	.	CLP020 EQU \$	
2092	6809	.	.	; COMPUTE NEW VECTOR PARAMETERS USING NEW	
2093	6809	.	.	; ENDPOINTS	
2094	6809	.	.	; TEST TO INSURE NEW COORDINATE ARE IN BOUNDS	
2095	6809	AF	.	XRA A ;RESET OCTANT	
2096	680A	32	D1 90	STA OCTANT	
2097	680D	2A	65 90	LHLD XFIN ;COMPUTE DELTA X	
2098	6810	CD	9F 65	CALL XCODE ;FIRST, SEE IF IN BOUNDS	
2099	6813	EB	.	XCHG ;DE = XFIN	
2100	6814	2A	69 90	LHLD XSTART ;HL = XSTART	
2101	6817	CD	9F 65	CALL XCODE ;SEE IF IN BOUNDS	
2102	681A	B7	.	ORA A ;TEST BOUNDS CODE	
2103	681B	C0	.	RNZ ;ONE IS OUT, DONT DRAW	
2104	681C	CD	8D 65	CALL DELXY ;COMPUTE DELTA X	
2105	681F	22	D6 90	SHLD DELTAX	
2106	6822	AF	.	XRA A ;CLEAR A FOR BOUNDS CODE	
2107	6823	2A	63 90	LHLD YFIN ;COMPUTE DELTA Y	
2108	6826	CD	8D 65	CALL YCODE ;FIRST, SEE IF IN BOUNDS	
2109	6829	EB	.	XCHG ;DE = YFIN	
2110	682A	2A	67 90	LHLD YSTART ;HL = YSTART	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 60
=====
2111      682D      CD  BD  65      CALL YCODE      ;SEE IF IN BOUNDS
2112      6830      B7  .   .      ORA  A          ;TEST THE BOUNDS CODE
2113      6831      C0  .   .      RNZ              ;ONE OF THEM IS OUT, EXIT
2114      6832      CD  8D  65      CALL DELXY      ;COMPUTE DELTA Y
2115      6835      22  D4  90      SHLD DELTAY
2116      6838      AF  .   .      XRA  A          ;SET Z FLAG
2117      6839      C9  .   .      RET
=====
```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 61
=====
2119     683A      . . .      ;*****
2120     683A      . . .      ; CLPALG--CLIPPING ALGORITHM
2121     683A      . . .      ; ENTRY HL = X COORD, DE = Y COORD
2122     683A      . . .      ;       XDEL, YDEL = DELTA
2123     683A      . . .      ;       B = BOUNDS CODE FOR POINT
2124     683A      . . .      ; EXIT HL = CLIPPED X, DE = CLIPPED Y
2125     683A      . . .      ;*****
2126     683A      . . .      CLPALG EQU $
2127     683A      22 59 90      SHLD XTEMP      ;STORE ENDPOINT COORDS
2128     683D      EB . .      XCHG
2129     683E      22 57 90      SHLD YTEMP
2130     6841      2A 5D 90      LHLD XDEL      ;FETCH DELTA X
2131     6844      22 55 90      SHLD TEMPDX    ;SAVE IT
2132     6847      2A 5B 90      LHLD YDEL
2133     684A      22 53 90      SHLD TEMPDY    ;SAVE DELTA Y
2134     684D      . . .      CPA005 EQU $
2135     684D      . . .      ; COMPUTE DELTA/2
2136     684D      2A 55 90      LHLD TEMPDX
2137     6850      CD 11 A3      CALL DIVHLR    ;DIVIDE BY 2 AND ROUND
2138     6853      22 55 90      SHLD TEMPDX
2139     6856      EB . .      XCHG          ;DE = DELX/2
2140     6857      2A 53 90      LHLD TEMPDY
2141     685A      CD 11 A3      CALL DIVHLR    ;DIVIDE BY 2 AND ROUND
2142     685D      22 53 90      SHLD TEMPDY    ;HL = DELY/2
2143     6860      . . .      ; IF BOTH DELTA X AND Y ARE 0, CLIPPING
2144     6860      . . .      ; IS DONE
2145     6860      7C . .      MOV A,H       ;TEST ALL REGISTERS
2146     6861      85 . .      ORA L
2147     6862      82 . .      ORA D
2148     6863      83 . .      ORA E
2149     6864      CA C9 68      JZ CPA040     ;DONE IF ALL ARE 0
2150     6867      . . .      ; ADD DELTA/2 TO ENDPOINT TO GET MIDPOINT
2151     6867      E5 . .      PUSH H       ;SAVE DELY/2
2152     6868      2A 59 90      LHLD XTEMP    ;ADD DELX/2 TO X
2153     686B      19 . .      DAD D
2154     686C      22 61 90      SHLD XMID     ;NOW HAVE NEW X MIDPOINT
2155     686F      D1 . .      POP D        ;RECALL DELY/2
2156     6870      2A 57 90      LHLD YTEMP
2157     6873      19 . .      DAD D        ;ADD DELY/2 TO Y
2158     6874      22 5F 90      SHLD YMID     ;HAVE NEW Y MIDPOINT
2159     6877      . . .      ; TEST TO SEE IF MIDPOINT IS OFF SCREEN
2160     6877      . . .      ; IF SO, USE MID POINT AS NEW ENDPOINT
2161     6877      . . .      ; IF NOT, KEEP CURRENT ENDPOINT
2162     6877      . . .      ; ONLY TEST THOSE EDGES THAT POINT IS BEYOND
2163     6877      . . .      ; SEE IF Y IS .LT. MIN + 1
2164     6877      EB . .      XCHG        ;DE = Y MIDPOINT
2165     6878      3E 04 .      MVI A,LTYMIN ;IS Y TOO SMALL?
2166     687A      A0 . .      ANA B
2167     687B      CA 88 68      JZ CPA010     ;NO, DONT TEST
2168     687E      2A 6E 90      LHLD YMIN     ;LOAD -1 * MIN VALUE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2169	6881	19	.	DAD D ;HL = Y - MIN	62
2170	6882	2B	.	DCX H ;**TEST AGAINST +MIN + 1**	
2171	6883	7C	.	MOV A,H ;IS HL -?	
2172	6884	B7	.	ORA A	
2173	6885	FA	BA 68	JM CPA030 ;YES, MIDPOINT IS OFF SCREEN	
2174	6888	.	.	; SEE IF Y IS .GT. MAX - 1	
2175	6888	.	.	CPA010 EQU \$	
2176	6888	3E	08	MVI A,GTMAX ;IS Y TOO LARGE?	
2177	688A	A0	.	ANA B	
2178	688B	CA	97 68	JZ CPA015 ;NO, DONT TEST	
2179	688E	2A	6C 90	LHLD YMAX ;LOAD -1 * MAX	
2180	6891	19	.	DAD D ;HL = Y - MAX	
2181	6892	7C	.	MOV A,H ;MID IS OFF IF Y IS .GE. MAX	
2182	6893	B7	.	ORA A ;IS HL +	
2183	6894	F2	BA 68	JP CPA030 ;YES, MIDPOINT IS OFF SCREEN	
2184	6897	.	.	; TEST X MIDPOINT	
2185	6897	.	.	; SEE IF X IS .LT. MIN + 1	
2186	6897	.	.	CPA015 EQU \$	
2187	6897	2A	61 90	LHLD XMID	
2188	689A	EB	.	XCHG ;DE = X MIDPOINT	
2189	689B	3E	01	MVI A,LTXMIN ;IS X TOO SMALL?	
2190	689D	A0	.	ANA B	
2191	689E	CA	AB 68	JZ CPA020 ;NO, DONT TEST	
2192	68A1	2A	72 90	LHLD XMIN ;LOAD -1 * MIN VALUE	
2193	68A4	19	.	DAD D ;HL = X - MIN	
2194	68A5	2B	.	DCX H ;**TEST AGAINST +MIN + 1**	
2195	68A6	7C	.	MOV A,H ;IS HL -?	
2196	68A7	B7	.	ORA A	
2197	68A8	FA	BA 68	JM CPA030 ;YES, MIDPOINT IS OFF SCREEN	
2198	68AB	.	.	; SEE IF X IS .GT. MAX - 1	
2199	68AB	.	.	CPA020 EQU \$	
2200	68AB	3E	02	MVI A,GTXMAX ;IS X TOO LARGE?	
2201	68AD	A0	.	ANA B	
2202	68AE	CA	4D 68	JZ CPA005 ;NO, GO THRU LOOP AGAIN	
2203	68B1	2A	70 90	LHLD XMAX ;LOAD -1 * MAX	
2204	68B4	19	.	DAD D ;HL = X - MAX	
2205	68B5	7C	.	MOV A,H ;MID IS OFF IF X IS .GE. MAX	
2206	68B6	B7	.	ORA A ;IS HL +?	
2207	68B7	.	.	; IF MIDPOINT IS ON SCREEN, IGNORE IT AND GO THRU	
2208	68B7	.	.	; LOOP AGAIN	
2209	68B7	FA	4D 68	JM CPA005 ;NO, MIDPOINT IS ON SCREEN	
2210	68BA	.	.	CPA030 EQU \$	
2211	68BA	.	.	; MIDPOINT IS OFF SCREEN, USE IT AS NEW ENDPOINT	
2212	68BA	2A	61 90	LHLD XMID	
2213	68BD	22	59 90	SHLD XTEMP ;X COORD	
2214	68C0	2A	5F 90	LHLD YMID	
2215	68C3	22	57 90	SHLD YTEMP ;Y COORD	
2216	68C6	C3	4D 68	JMP CPA005 ;GO THRU LOOP AGAIN	
2217	68C9	.	.	CPA040 EQU \$	
2218	68C9	.	.	; EXIT WITH HL = X AND DE = Y	

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS      PAGE 63
=====
```

2219	68C9	2A 57 90	LHLD YTEMP	
2220	68CC	EB . .	XCHG	
2221	68CD	2A 59 90	LHLD XTEMP	
2222	68D0	C9 . .	RET	

```
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2224	68D1	.	.	;*****	64
2225	68D1	.	.	; TEKSEQ--ESC * T RECEIVED, SET UP FOR TEKTRONIX	
2226	68D1	.	.	; MODE CONTROL ESCAPE SEQUENCE	
2227	68D1	.	.	;*****	
2228	68D1	.	.	TEKSEQ EQU \$	
2229	68D1	21	4E 61	LXI H,TEKTAB ;SET NEW RANGE TABLE	
2230	68D4	C3	86 63	JMP SETRTB	
2231	68D7	.	.	;*****	
2232	68D7	.	.	; TKOFF--TURN TEK MODE OFF	
2233	68D7	.	.	; TURNS MONITOR MODE OFF	
2234	68D7	.	.	;*****	
2235	68D7	.	.	TKOFF EQU \$	
2236	68D7	3E	41 .	MVI A,SCLD+UNSCLD	
2237	68D9	C3	53 A2	JMP CLTKFL ;CLEAR TEK FLAGS	
2238	68DC	.	.	;*****	
2239	68DC	.	.	; TKSCLD--TURN SCALED TEK MODE ON	
2240	68DC	.	.	; TURNS MONITOR MODE ON	
2241	68DC	.	.	;*****	
2242	68DC	.	.	TKSCLD EQU \$	
2243	68DC	3E	40 .	MVI A,SCLD	
2244	68DE	CD	4D A2	CALL STTKFL ;TURN SCALED ON	
2245	68E1	E6	C4 .	ANI -1-UNSCLD-SUPCHR-GINMOD-GSMODE-MARG1	
2246	68E3	77	. .	MOV M,A	
2247	68E4	CD	F7 75	CALL GTXON1 ;TURN GRAPHICS TEXT ON	
2248	68E7	.	.	; SET FIXED TEXT PARAMETERS	
2249	68E7	AF	. .	XRA A ;SIZE = 1X	
2250	68E8	32	DA FB	STA TXMAG	
2251	68EB	3E	01 .	MVI A,SLANT ;SLANT OFF	
2252	68ED	CD	60 A2	CALL CLFLG6	
2253	68F0	E6	87 .	ANI -1-RTJUST-CNTR-TOPCH-MIDCH	
2254	68F2	77	. .	MOV M,A ;LORG = 0	
2255	68F3	AF	. .	XRA A ;UPRIGHT CHARACTERS	
2256	68F4	C3	48 76	JMP ANGLE	
2257	68F7	.	.	;*****	
2258	68F7	.	.	; TKUNSC--TURN UNSCALED TEK MODE ON	
2259	68F7	.	.	; TURNS MONITOR MODE ON	
2260	68F7	.	.	;*****	
2261	68F7	.	.	TKUNSC EQU \$	
2262	68F7	3E	01 .	MVI A,UNSCLD	
2263	68F9	CD	4D A2	CALL STTKFL ;TURN UNSCALED ON	
2264	68FC	E6	85 .	ANI -1-SCLD-SUPCHR-GINMOD-GSMODE-MARG1	
2265	68FE	77	. .	MOV M,A	
2266	68FF	C9	. .	RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 65
=====
2268      6900      . . .      ;*****
2269      6900      . . .      ; TRMSTP--SET GIN MODE TERMINATOR STRAP
2270      6900      . . .      ; 0 = CR
2271      6900      . . .      ; 1 = CR, EOT
2272      6900      . . .      ; 2 = NOTHING
2273      6900      . . .      ;*****
2274      6900      . . .      TRMSTP EQU $
2275      6900      3E 02      MVI A,2          ;MAX VALUE
2276      6902      CD 43 6D      CALL GETPRM      ;GET SINGLE PARAM
2277      6905      C2 C1 99      JNZ GEXIT        ;IGNORE IF BAD
2278      6908      32 DC FB      STA TEKTRM       ;STORE TERMINATOR
2279      690B      C3 C1 99      JMP GEXIT
2280      690E      . . .      ;*****
2281      690E      . . .      ; SETBRK--SET PAGE FULL BREAK FLAG
2282      690E      . . .      ;*****
2283      690E      . . .      SETBRK EQU $
2284      690E      0E 01      MVI C,PFBRK
2285      6910      . . .      SETPF1 EQU $
2286      6910      C5 . .      PUSH B           ;SAVE STRAP
2287      6911      3E 01      MVI A,1          ;MAX VALUE
2288      6913      CD 43 6D      CALL GETPRM      ;GET THE PARAMETER
2289      6916      C1 . .      POP B
2290      6917      C2 C1 99      JNZ GEXIT        ;IGNORE IF BAD
2291      691A      21 C4 FB      LXI H,TEKPF      ;ADDRESS OF STRAP FLAGS
2292      691D      3D . .      DCR A           ;WAS PARAMETER 0?
2293      691E      79 . .      MOV A,C          ;(RECALL THE FLAG)
2294      691F      FA 27 69      JM SPFU10        ;YES, CLEAR THE FLAG
2295      6922      B6 . .      ORA M           ;SET THE STRAP
2296      6923      77 . .      MOV M,A
2297      6924      C3 C1 99      JMP GEXIT
2298      6927      . . .      SPFU10 EQU $
2299      6927      . . .      ; CLEAR THE STRAP
2300      6927      2F . .      CMA
2301      6928      A6 . .      ANA M
2302      6929      77 . .      MOV M,A
2303      692A      C3 C1 99      JMP GEXIT
2304      692D      . . .      ;*****
2305      692D      . . .      ; SETBSY--SET PAGE FULL BUSY STRAP
2306      692D      . . .      ;*****
2307      692D      . . .      SETBSY EQU $
2308      692D      0E 02      MVI C,PFBUSY
2309      692F      C3 10 69      JMP SETPF1
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 66
2311	6932	.	.	*****	
2312	6932	.	.	; TKSTRP--SET KEYBOARD STRAPS FOR TEK MODE	
2313	6932	.	.	; P => TURN SCALED ON, Q => TURN UNSCALED ON	
2314	6932	.	.	; IF MODE ALREADY SET, DONT SET IT AGAIN	
2315	6932	.	.	; AS SETTING MODE CHANGES TEXT SIZE, ETC.	
2316	6932	.	.	; ENTRY--DONT CARE	
2317	6932	.	.	; EXIT---ALL DESTROYED	
2318	6932	.	.	*****	
2319	6932	.	.	TKSTRP EQU \$	
2320	6932	3A	AD 90	LDA TKFLGS ;FETCH CURRENT STATE	
2321	6935	4F	.	MOV C,A	
2322	6936	3A	FA FF	LDA KBJMP2 ;GET JUMPERS	
2323	6939	E6	60 .	ANI PJMPR+QJMPR ;P OR Q OUT?	
2324	693B	CA	D7 68	JZ TKOFF ;NO, TURN TEK MODE OFF	
2325	693E	FE	60 .	CPI PJMPR+QJMPR ;BOTH OUT?	
2326	6940	CA	D7 68	JZ TKOFF ;YES, TURN TEK MODE OFF	
2327	6943	FE	20 .	CPI PJMPR ;JUST P OUT?	
2328	6945	CA	4F 69	JZ TCH010 ;YRS, TURN SCALED MODE ON	
2329	6948	.	.	; TURN UNSCALED MODE ON IF NOT ALREADY SO	
2330	6948	3E	01 .	MVI A,UNSCLD ;UNSCALED ALREADY ON?	
2331	694A	A1	.	ANA C	
2332	694B	CA	F7 68	JZ TKUNSC ;NO, TURN IT ON	
2333	694E	C9	.	RET ;YES, DONE	
2334	694F	.	.	TCH010 EQU \$	
2335	694F	.	.	; TURN SCALED MODE ON IF NOT ALREADY ON	
2336	694F	3E	40 .	MVI A,SCLD ;SCALED ALREADY ON?	
2337	6951	A1	.	ANA C	
2338	6952	CA	DC 68	JZ TKSCLD ;NO, TURN IT ON	
2339	6955	C9	.	RET ;YES, DONE	
2340	6956	.	.	*****	
2341	6956	.	.	; CHKTEK--SEE IF IN EITHER TEK MODE	
2342	6956	.	.	; EXIT NZ => IN TEK MODE, A DESTROYED	
2343	6956	.	.	*****	
2344	6956	.	.	CHKTEK EQU \$	
2345	6956	3A	AD 90	LDA TKFLGS	
2346	6959	E6	41 .	ANI SCLD+UNSCLD ;CHECK BOTH MODES	
2347	695B	C9	.	RET	
2348	695C	.	.	*****	
2349	695C	.	.	; CKSCLD--SEE IF IN SCALED TEK MODE	
2350	695C	.	.	; EXIT NZ => IN SCALED TEK MODE	
2351	695C	.	.	; A DESTROYED	
2352	695C	.	.	*****	
2353	695C	.	.	CKSCLD EQU \$	
2354	695C	3A	AD 90	LDA TKFLGS	
2355	695F	E6	40 .	ANI SCLD	
2356	6961	C9	.	RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 67
=====
2358      6962      . . .      ;*****
2359      6962      . . .      ; TKSTUP--GS RECEIVED, SET UP FOR TEK VECTORS
2360      6962      . . .      ; ENTRY--DONT CARE
2361      6962      . . .      ; EXIT---ALL REGISTERS DESTROYED
2362      6962      . . .      ;*****
2363      6962      . . .      TKSTUP EQU $
2364      6962      CD 56 69      CALL CHKTEK      ;IN TEK MODE?
2365      6965      C8 . .      RZ                ;NO, IGNORE THE GS
2366      6966      21 6C 61      LXI H,TKGSTB    ;SET NEW RANGE TABLE
2367      6969      CD 86 63      CALL SETRTB
2368      696C      3E 01 .      MVI A,MOVE      ;DO A MOVE AFTER GS
2369      696E      CD 26 A2      CALL STFLG1
2370      6971      3E 34 .      MVI A,XNEXT+SUPCHR+GINMOD ;CLEAR FLAGS
2371      6973      CD 53 A2      CALL CLTKFL
2372      6976      F6 08 .      ORI GSMODE      ;SET TEK GRAPHICS MODE
2373      6978      77 . .      MOV M,A
2374      6979      3E 01 .      MVI A,SUPRO     ;SUPRESS CURSOR
2375      697B      C3 8A 9E      JMP SUPRGC
2376      697E      . . .      ;*****
2377      697E      . . .      ; GSEND--US RECEIVED, TERMINATE TEK GRAPHICS MODE
2378      697E      . . .      ;*****
2379      697E      . . .      GSEND EQU $
2380      697E      CD A9 6B      CALL CLRSP      ;CLEAR TEXT SUPRESS
2381      6981      CD AB 9E      CALL ENABO      ;RE-ENABLE THE CURSOR
2382      6984      C3 4F 00      JMP ZESCND      ;BACK TO ALPHA MODE
2383      6987      . . .      ;*****
2384      6987      . . .      ; TEK VECTOR PARAMETER BYTES RECEIVED
2385      6987      . . .      ;*****
2386      6987      . . .      HIXY EQU $      ;EITHER HI X OR Y BYTE
2387      6987      3A AD 90      LDA TKFLGS      ;IS THIS THE X BYTE?
2388      698A      E6 04 .      ANI XNEXT
2389      698C      C2 98 69      JNZ HXY010      ;YES
2390      698F      . . .      ;BYTE IS HI Y PARAMETER
2391      698F      3A 88 FF      LDA ZCHAR      ;FETCH PARAMETER
2392      6992      E6 1F .      ANI 37Q        ;DELETE TAG BITS
2393      6994      32 A9 90      STA YHI
2394      6997      C9 . .      RET
2395      6998      . . .      ;BYTE IS HI X PARAMETER
2396      6998      . . .      HXY010 EQU $
2397      6998      3A 88 FF      LDA ZCHAR
2398      6998      E6 1F .      ANI 37Q        ;DELETE TAG BITS
2399      699D      32 AB 90      STA XHI
2400      69A0      C9 . .      RET
2401      69A1      . . .      ;
2402      69A1      . . .      ;
2403      69A1      . . .      LOWY EQU $
2404      69A1      3A 88 FF      LDA ZCHAR
2405      69A4      E6 1F .      ANI 37Q        ;DELETE TAG BITS
2406      69A6      32 AA 90      STA YLOW
2407      69A9      . . .      ; AFTER LOW Y RECEIVED, NEXT IS HI X
=====

```

13255
2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS          PAGE 68
=====
2408     69A9     3E 04 .      MVI A,XNEXT ;SET X NEXT FLAG
2409     69AB     C3 4D A2     JMP STTKFL
=====
```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 69
=====
2411      69AE      .      .      .      ;*****
2412      69AE      .      .      .      ; LOWX--HAVE LOW X BYTE, DRAW THE VECTOR
2413      69AE      .      .      .      ;*****
2414      69AE      .      .      .      LOWX      EQU      $
2415      69AE      3A      88      FF      LDA      ZCHAR      ;FETCH BYTE
2416      69B1      E6      1F      .      ANI      370        ;DELETE TAG BITS
2417      69B3      32      AC      90      STA      XLOW
2418      69B6      3E      04      .      MVI      A,XNEXT    ;CLEAR HI X FLAG
2419      69B8      CD      53      A2      CALL     CLTKFL
2420      69BB      .      .      .      ; PUT VECTOR INTO PROPER FORMAT--MERGE THE BYTES
2421      69BB      21      A9      90      LXI      H,YHI      ;REFORMAT Y BYTES
2422      69BE      CD      D1      A2      CALL     FORMAT
2423      69C1      22      D8      90      SHLD     YNEW
2424      69C4      21      AB      90      LXI      H,XHI      ;REFORMAT X BYTES
2425      69C7      CD      D1      A2      CALL     FORMAT
2426      69CA      22      DA      90      SHLD     XNEW
2427      69CD      .      .      .      ;IF SCALED MODE SELECTED, DO THE SCALING
2428      69CD      .      .      .      ;DIVIDE X BY 2
2429      69CD      .      .      .      ;MULTIPLY Y BY 59, AND DIVIDE BY 128
2430      69CD      CD      5C      69      CALL     CKSCLD    ;IN SCALED TEK MODE?
2431      69D0      CA      FB      69      JZ       LWX005    ;NO, PROCES UNSCALED X
2432      69D3      .      .      .      ;DIVIDE X BY 2
2433      69D3      CD      1A      A3      CALL     DIVHL1    ;HL = HL/2
2434      69D6      22      DA      90      SHLD     XNEW
2435      69D9      .      .      .      ;MULTIPLY Y BY 59
2436      69D9      2A      D8      90      LHLD     YNEW
2437      69DC      5D      .      .      MOV      E,L
2438      69DD      54      .      .      MOV      D,H        ;HL=DE=YNEW
2439      69DE      29      .      .      DAD      H          ;2*Y
2440      69DF      29      .      .      DAD      H          ;4*Y
2441      69E0      19      .      .      DAD      D          ;5*Y
2442      69E1      29      .      .      DAD      H          ;10*Y
2443      69E2      29      .      .      DAD      H          ;20*Y
2444      69E3      4D      .      .      MOV      C,L
2445      69E4      44      .      .      MOV      B,H        ;BC = 20 * Y
2446      69E5      29      .      .      DAD      H          ;40*Y
2447      69E6      09      .      .      DAD      B          ;60*Y
2448      69E7      EB      .      .      XCHG                    ;HL = Y, DE=60*Y
2449      69E8      CD      09      A3      CALL     NEGATE    ;HL = -Y
2450      69EB      19      .      .      DAD      D          ;HL = 59 * Y
2451      69EC      .      .      .      ;DIVIDE 59*Y BY 128 (SHIFT RIGHT 7 TIMES)
2452      69EC      0E      07      .      MVI      C,7        ;DIVIDE BY 128
2453      69EE      CD      44      6C      CALL     DIV128    ;HL = HL / 128
2454      69F1      7C      .      .      MOV      A,H        ;WANT 9 LSB ONLY
2455      69F2      E6      01      .      ANI      10
2456      69F4      67      .      .      MOV      H,A
2457      69F5      22      D8      90      SHLD     YNEW
2458      69F8      C3      DB      65      JMP      VECTOR
2459      69FB      .      .      .      LWX005 EQU      $
2460      69FB      .      .      .      ; UNSCALED TEK COORD--SUBTRACT RELOC ORIGIN
=====

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 70
=====
2461      69FB      . . .      ; FROM COORDINATE VALUES
2462      69FB      EB . .      XCHG ;DE = X COORD
2463      69FC      2A 9A 90    LHLD XORG ;RELOC ORG, X COORD
2464      69FF      CD 09 A3    CALL NEGATE ; -X ORG CUORD
2465      6A02      19 . .      DAD D ;HL = X - RELOC ORG
2466      6A03      22 DA 90    SHLD XNEW
2467      6A06      2A D8 90    LHLD YNEW
2468      6A09      EB . .      XCHG ;DE = Y COORD
2469      6A0A      2A 98 90    LHLD YORG
2470      6A0D      CD 09 A3    CALL NEGATE ;HL = - ORG Y COORD
2471      6A10      19 . .      DAD D
2472      6A11      22 D8 90    SHLD YNEW
2473      6A14      C3 DB 65    JMP VECTOR ;DRAW THE VECTOR
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2475	6A17	.	.	*****	71
2476	6A17	.	.	; TEKESC-ESCAPE RECEIVED IN GMODE	
2477	6A17	.	.	*****	
2478	6A17	.	.	TEKESC EQU \$	
2479	6A17	21	8C 61	LXI H, TESCTB ;SET NEW RANGE TABLE	
2480	6A1A	C3	86 63	JMP SETRTB	
2481	6A1D	.	.	*****	
2482	6A1D	.	.	; PAGE--CLEAR SCREEN AND RETURN TO ALPHA MODE	
2483	6A1D	.	.	; ENTRY--DONT CARE	
2484	6A1D	.	.	; EXIT---ALL REGISTERS DESTROYED	
2485	6A1D	.	.	*****	
2486	6A1D	.	.	PAGE EQU \$	
2487	6A1D	CD	56 69	CALL CHKTEK ;IN TEK MODE?	
2488	6A20	CA	4F 00	JZ ZESCND ;NO, IGNORE	
2489	6A23	3E	3A .	MVI A, SUPCHR+GINMOD+GSMODE+MARG1	
2490	6A25	CD	53 A2	CALL CLTKFL ;CLEAR FLAGS	
2491	6A28	CD	AB 9E	CALL ENABO ;RE-ENABLE THE CURSOR	
2492	6A2B	.	.	; HOME CURSOR IF SCALED TEK MODE	
2493	6A2B	CD	50 6A	CALL TEKHOM	
2494	6A2E	CD	78 6D	CALL GCLR1 ;CLEAR THE SCREEN	
2495	6A31	C3	4F 00	JMP ZESCND ;RETURN TO ALPHA	
2496	6A34	.	.	*****	
2497	6A34	.	.	; TEKRPT--INVALID CHAR AFTER ESCAPE IN TEK	
2498	6A34	.	.	; GRAPHICS MODE. REPEAT LAST CHAR USING TEK	
2499	6A34	.	.	; GRAPHICS RANGE TABLE	
2500	6A34	.	.	*****	
2501	6A34	.	.	TEKRPT EQU \$	
2502	6A34	21	6C 61	LXI H, TKGSTB ;RESTORE GRAPHICS TABLE	
2503	6A37	.	.	TKRPT1 EQU \$	
2504	6A37	CD	86 63	CALL SETRTB	
2505	6A3A	21	88 FF	LXI H, ZCHAR ;RECALL CHAR	
2506	6A3D	4E	.	MOV C, M ;LEAVE IN C	
2507	6A3E	CD	82 00	CALL ZCHINT ;USE CHINT TO PROCESS	
2508	6A41	21	D1 FF	LXI H, ZESCFG ;RESET ESC SEQ COUNTER	
2509	6A44	36	FF .	MVI M, -1	
2510	6A46	C9	.	RET	
2511	6A47	.	.	*****	
2512	6A47	.	.	; TEKCR--CARRIAGE RETURN RECEIVED IN TEK	
2513	6A47	.	.	; GRAPHICS MODE. TERMINATE GRAPHICS AND DO A RET	
2514	6A47	.	.	*****	
2515	6A47	.	.	TEKCR EQU \$	
2516	6A47	CD	4F 00	CALL ZESCND ;TERMINATE GRAPHICS	
2517	6A4A	CD	AB 9E	CALL ENABO ;RE-ENABLE THE CURSOR	
2518	6A4D	C3	C0 00	JMP ZCRRET ;DO A RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 72
=====
2520      6A50      . . .      ;*****
2521      6A50      . . .      ; TEKHOM--HOME TEK CURSOR TO PROPER MARGIN IF
2522      6A50      . . .      ; IN SCALED MODE
2523      6A50      . . .      ;*****
2524      6A50      . . .      TEKHOM EQU $
2525      6A50      3A AD 90      LDA TKFLGS      ;IN SCALED MODE?
2526      6A53      4F . .      MOV C,A        ;(SAVE IN C)
2527      6A54      E6 40 .      ANI SCLD
2528      6A56      C8 . .      RZ              ;NO, DONE
2529      6A57      21 00 00      LXI H,XMARG0   ;ASSUME AT MARGIN 0
2530      6A5A      3E 02 .      MVI A,MARG1    ;REALLY THERE?
2531      6A5C      A1 . .      ANA C
2532      6A5D      CA 63 6A      JZ TKH010      ;YES
2533      6A60      21 03 01      LXI H,XMARG1   ;NO, SET FOR MARGIN 1
2534      6A63      . . .      TKH010 EQU $
2535      6A63      22 DA 90      SHLD XNEW      ;STORE X COORD
2536      6A66      21 5E 01      LXI H,YTEKHM   ;SET Y TO HOME
2537      6A69      22 D8 90      SHLD YNEW
2538      6A6C      C3 85 78      JMP PCH1       ;UPDATE CURRENT POINT
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
2540	6A6F	.	.	*****
2541	6A6F	.	.	; TEKAC--ESC ENQ RECEIVED IN A/N MODE, PREPARE
2542	6A6F	.	.	; TO SEND CURRENT POINT = A/N CURSOR POSITION
2543	6A6F	.	.	; AND STATUS
2544	6A6F	.	.	; STATUS BITS
2545	6A6F	.	.	; BIT 5 = 1 IF NO HARD COPY
2546	6A6F	.	.	; BIT 4 = 0 (NOT IN GRAPHICS)
2547	6A6F	.	.	; BIT 3 = 1 (NOT IN GRAPHICS)
2548	6A6F	.	.	; BIT 2 = 1 IF AT MARGIN 1
2549	6A6F	.	.	; BIT 1 = 1 NO AUX DEVICE
2550	6A6F	.	.	; ENTRY--DONT CARE
2551	6A6F	.	.	; EXIT--ALL REGISTERS DESTROYED
2552	6A6F	.	.	*****
2553	6A6F	.	.	TEKAC EQU \$
2554	6A6F	CD	D2 00	CALL ZCKRMT ;IN REMOTE?
2555	6A72	C4	56 69	CNZ CHKTEK ;AND IN TEK MODE?
2556	6A75	CA	4F 00	JZ ZESCND ;NO, IGNORE
2557	6A78	3E	81 .	MVI A,ACBLOK ;SET STATUS BLOCK #
2558	6A7A	.	.	TEKST1 EQU \$
2559	6A7A	32	6B 90	STA GSBLOK
2560	6A7D	01	00 00	LXI B,0 ;SET BLOCK XFER FLAGS
2561	6A80	CD	D5 00	CALL ZSBXFR
2562	6A83	C3	4F 00	JMP ZESCND
2563	6A86	.	.	*****
2564	6A86	.	.	; TKACGO--BLOCK XFER TRIGGER RECEIVED, SEND
2565	6A86	.	.	; A/N CURSOR AND STATUS
2566	6A86	.	.	*****
2567	6A86	.	.	TKACGO EQU \$
2568	6A86	CD	98 6A	CALL GTKST ;GET TEK STATUS WORD
2569	6A89	F6	25 .	ORI 45Q ;SET OTHER FIXED FLAGS
2570	6A8B	CD	60 6C	CALL XPUTDC ;SEND STATUS
2571	6A8E	.	.	; SEND CURRENT POINT = A/N CURSOR POSITION
2572	6A8E	2A	DC 90	LHLD YCURR
2573	6A91	EB	.	XCHG
2574	6A92	2A	DE 90	LHLD XCURR
2575	6A95	C3	B2 6B	JMP SNDTEK ;SEND X,Y
2576	6A98	.	.	*****
2577	6A98	.	.	; GTKST--GET TEK MODE STATUS BYTE
2578	6A98	.	.	; EXIT A = STATUS
2579	6A98	.	.	*****
2580	6A98	.	.	GTKST EQU \$
2581	6A98	0E	10 .	MVI C,20Q ;ASSUME NO PRINTER
2582	6A9A	CD	A9 6A	CALL CHKPTR ;SEE IF ONE THERE
2583	6A9D	CA	A2 6A	JZ GTS010 ;NO THERES NOT
2584	6AA0	0E	00 .	MVI C,0 ;YES, SET PRINTER PRESENT FL
2585	6AA2	.	.	GTS010 EQU \$
2586	6AA2	3A	AD 90	LDA TKFLGS ;GET MARGIN 1 BIT
2587	6AA5	E6	02 .	ANI MARG1
2588	6AA7	B1	.	ORA C ;SET PRINTER FLAG
2589	6AA8	C9	.	RET

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 74
=====
2591      6AA9      . . .      ;*****
2592      6AA9      . . .      ; CHKPTR--SEE IF NON RS232 PRINTER PRESENT
2593      6AA9      . . .      ; EXIT NZ => PRINTER PRESENT
2594      6AA9      . . .      ;      A DESTROYED
2595      6AA9      . . .      ;*****
2596      6AA9      . . .      CHKPTR EQU $
2597      6AA9      3A 77 FE      LDA PTRFLG      ;ANY PRINTER AT ALL?
2598      6AAC      B7 . .      ORA A
2599      6AAD      C8 . .      RZ              ;NO
2600      6AAE      FE 02 .      CPI 2Q          ;IS IT AN RS232 OR NOT?
2601      6AB0      C9 . .      RET            ;Z SET IF ITS AN RS232
2602      6AB1      . . .      ;*****
2603      6AB1      . . .      ; TEKCP--ESC ENQ RECEIVED IN GRAPHICS MODE,
2604      6AB1      . . .      ; PREPARE TO SEND CURRENT TEK POINT AND STATUS
2605      6AB1      . . .      ; STATUS BITS SAME EXCEPT
2606      6AB1      . . .      ; BIT 4 = 1 IN GRAPHICS
2607      6AB1      . . .      ; BIT 3 = 0 IN GRAPHICS
2608      6AB1      . . .      ;*****
2609      6AB1      . . .      TEKCP EQU $
2610      6AB1      CD D2 00      CALL ZCKRMT     ;IN REMOTE
2611      6AB4      C4 56 69      CNZ CHKTEK     ;AND IN TEK MODE?
2612      6AB7      CA 4F 00      JZ ZESCND      ;NO, IGNORE
2613      6ABA      3E 82 .      MVI A,CPBLOK  ;SET STATUS BLOCK #
2614      6ABC      C3 7A 6A      JMP TEKST1
2615      6ABF      . . .      ;*****
2616      6ABF      . . .      ; TKCPGO--BLOCK TRIGGER RECEIVED, SEND CURRENT
2617      6ABF      . . .      ; POINT AND STATUS
2618      6ABF      . . .      ;*****
2619      6ABF      . . .      TKCPGO EQU $
2620      6ABF      CD 98 6A      CALL GTKST     ;GET STATUS BITS
2621      6AC2      F6 29 .      ORI 51Q        ;SET FIXED BITS
2622      6AC4      CD 60 6C      CALL XPUTDC    ;SEND THEM
2623      6AC7      . . .      ; SEND CURRENT TEK POINT
2624      6AC7      21 A9 90      LXI H,YHI
2625      6ACA      CD D1 A2      CALL FORMAT    ;CONVERT TO PROPER FORMAT
2626      6ACD      E5 . .      PUSH H
2627      6ACE      21 AB 90      LXI H,XHI
2628      6AD1      CD D1 A2      CALL FORMAT
2629      6AD4      D1 . .      POP D          ;RECALL Y
2630      6AD5      CD 82 6B      CALL SNDTEK    ;SEND X,Y
2631      6AD8      21 6C 61      LXI H,TKGSTB  ;RESTORE GS TABLE
2632      6ADB      C3 86 63      JMP SETRTB
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
2634      6ADE      .      .      .      ;*****
2635      6ADE      .      .      .      ; STGIN--ESC SUB RECEIVED, START TEK GIN MODE
2636      6ADE      .      .      .      ; ENTRY--DONT CARE
2637      6ADE      .      .      .      ; EXIT---ALL REGISTERS DESTROYED
2638      6ADE      .      .      .      ;*****
2639      6ADE      .      .      .      STGIN EQU $
2640      6ADE      CD D2 00      CALL ZCKRMT      ;IN REMOTE?
2641      6AE1      C4 56 69      CNZ  CHKTEK      ;AND IN TEK MODE?
2642      6AE4      CA 4F 00      JZ   ZESCND      ;NO, IGNORE
2643      6AE7      21 A0 61      LXI H,GINTAB    ;SET RANGE TABLE
2644      6AEA      .      .      .      STGIN1 EQU $
2645      6AEA      CD 86 63      CALL SETRTB
2646      6AED      3E 08 .      MVI A,GSMODE    ;CLEAR GRAPHICS MODE
2647      6AEF      CD 53 A2      CALL CLTKFL
2648      6AF2      F6 10 .      ORI  GINMOD     ;SET GIN MODE
2649      6AF4      77 . . .      MOV  M,A
2650      6AF5      .      .      .      ; PUT CURSOR TO WHERE IT WAS WHEN THE LAST
2651      6AF5      .      .      .      ; POINT WAS DIGITIZED
2652      6AF5      3A 97 90      LDA  GFLGS6     ;IN GTEXT MODE
2653      6AF8      E6 02 .      ANI  GTEXT
2654      6AFA      CA D6 6F      JZ   TGCON1     ;NO, JUST TURN CURSOR ON
2655      6AFD      CD 02 70      CALL TGCOF1     ;TURN THE CURSOR OFF
2656      6B00      2A D6 FB      LHLD XGINSV     ;MOVE CURSOR TO LAST POINT
2657      6B03      22 CF 90      SHLD NEWGCX     ;DIGITIZED
2658      6B06      2A D4 FB      LHLD YGINSV
2659      6B09      22 CD 90      SHLD NEWGCY
2660      6B0C      3E 21 .      MVI A,GCM1+GCM3 ;SET CURSOR MOVED FLAGS
2661      6B0E      CD 40 A2      CALL STFLG5
2662      6B11      C3 D6 6F      JMP  TGCON1     ;TURN THE CURSOR ON
2663      6B14      .      .      .      ;*****
2664      6B14      .      .      .      ; GINCH--CHARACTER RECEIVED IN GIN MODE
2665      6B14      .      .      .      ; IF FROM KEYBOARD, SEND IT AND END GIN
2666      6B14      .      .      .      ; IF FROM COMPUTER, IGNORE
2667      6B14      .      .      .      ; CHAR NOT ACTUALLY SENT UNTIL BLOCK XFER
2668      6B14      .      .      .      ; TRIGGER RECEIVED
2669      6B14      .      .      .      ;*****
2670      6B14      .      .      .      GINCH EQU $
2671      6B14      CD C3 00      CALL ZDCIO      ;FROM KEYBOARD?
2672      6B17      C2 F0 00      JNZ  ZCKCTL     ;CHECK FOR BLOCK TRIGGER
2673      6B1A      .      .      .      GINCH1 EQU $
2674      6B1A      .      .      .      ; SET BLOCK XFER FLAGS
2675      6B1A      3E 83 .      MVI A,GINBLK
2676      6B1C      .      .      .      GINCH2 EQU $
2677      6B1C      32 6B 90      STA  GSBLOK
2678      6B1F      01 00 00      LXI  B,0
2679      6B22      CD 5B 00      CALL ZSBXFA
2680      6B25      .      .      .      ; SAVE CURSOR POSITION
2681      6B25      2A CF 90      LHLD NEWGCX
2682      6B28      22 D6 FB      SHLD XGINSV
2683      6B2B      2A CD 90      LHLD NEWGCY
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 76
=====
2684      6B2E      22  D4  FB          SHLD YGINSV
2685      6B31      .   .   .          ; USE CHAR AS GIN CHAR
2686      6B31      3A  88  FF          LDA  ZCHAR      ;FETCH THE CHAR
2687      6B34      32  E2  FA          STA  GINCHR     ;SAVE IT
2688      6B37      .   .   .          GINCH3 EQU $
2689      6B37      .   .   .          ; TERMINATE GIN MODE
2690      6B37      3E  10  .          MVI  A,GINMOD  ;CLEAR GIN
2691      6B39      CD  53  A2         CALL CLTKFL
2692      6B3C      CD  02  70         CALL TGCUF1    ;TURN CURSOR OFF
2693      6B3F      C3  4F  00         JMP  ZESCND    ;RESTORE RANGE TABLES
=====
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
2695	6B42	.	.	*****
2696	6B42	.	.	; TKGNGO--SEND GIN CHAR AND CURSOR COORDS
2697	6B42	.	.	*****
2698	6B42	.	.	TKGNGO EQU \$
2699	6B42	3A	E2 FA	LDA GINCHR
2700	6B45	CD	60 6C	CALL XPUTDC ;SEND THE CHAR
2701	6B48	.	.	; SEND CURSOR LOCATION
2702	6B48	.	.	TKGCGO EQU \$
2703	6B48	2A	D4 FB	LHLD YGINSV ;Y COORD
2704	6B4B	22	D8 90	SHLD YNEW ;MOVE A/N CURSOR THERE
2705	6B4E	EB	.	XCHG
2706	6B4F	2A	D6 FB	LHLD XGINSV ;X COORD
2707	6B52	22	DA 90	SHLD XNEW
2708	6B55	CD	B2 6B	CALL SNDTEK ;SEND X,Y
2709	6B58	C3	32 98	JMP CPUPDA ;PUT A/N CURSOR THERE]
2710	6B5B	.	.	*****
2711	6B5B	.	.	; GINEND--ESC ENQ RECEIVED IN GIN MODE, SEND
2712	6B5B	.	.	; CURSOR LOCATION IMMEDIATELY
2713	6B5B	.	.	*****
2714	6B5B	.	.	GINEND EQU \$
2715	6B5B	3E	84 .	MVI A,GCBLK ;SET XFER PENDING FLAGS
2716	6B5D	32	6B 90	STA GSBLOK
2717	6B60	01	00 00	LXI B,0
2718	6B63	CD	5B 00	CALL ZSBXFA
2719	6B66	C3	4F 00	JMP ZESCND ;END ESCAPE SEQUENCE

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2721	6B69	.	.	;*****	78
2722	6B69	.	.	;GINESC--ESCAPE RECEIVED IN GIN MODE	
2723	6B69	.	.	;DO TEST FOR 2 CHAR ESCAPE SEQUENCE AS GENERATED	
2724	6B69	.	.	;BY KEYBOARD FUNCTION KEY. IF SO, IGNORE THE KEY	
2725	6B69	.	.	;*****	
2726	6B69	.	.	GINESC EQU \$	
2727	6B69	CD	C3 00	CALL ZDCIO ;FROM KEYBOARD?	
2728	6B6C	C2	7B 6B	JNZ GNE010 ;NO, USE NEW RANGE TABLE	
2729	6B6F	.	.	;IF ESC COMES FROM LOCAL FUNCTION KEY, IGNORE	
2730	6B6F	.	.	;OTHERWISE, USE ESCAPE AS GIN CHAR	
2731	6B6F	21	9C FF	LXI H,ZCHRIN ;FETCH ACTUAL INPUT CHAR	
2732	6B72	7E	.	MOV A,M	
2733	6B73	B7	.	ORA A ;2 CHAR SEQ?	
2734	6B74	F2	1A 6B	JP GINCH1 ;NO, USE ESC AS GIN CHAR	
2735	6B77	E6	7F .	ANI 177Q ;YES, SET TO IGNORE NEXT CHA	
2736	6B79	77	.	MOV M,A	
2737	6B7A	C9	.	RET	
2738	6B7B	.	.	GNE010 EQU \$	
2739	6B7B	21	AC 61	LXI H,GNECTB ;NO, SET NEW RANGE TABLE	
2740	6B7E	C3	86 63	JMP SETRTB	
2741	6B81	.	.	;*****	
2742	6B81	.	.	;ESCCH--PROCESS CHAR RECEIVED AFTER ESCAPE	
2743	6B81	.	.	;WHILE IN GIN MODE	
2744	6B81	.	.	;*****	
2745	6B81	.	.	ESCCH EQU \$	
2746	6B81	CD	C3 00	CALL ZDCIO ;FROM KEYBOARD?	
2747	6B84	CA	1A 6B	JZ GINCH1 ;YES, PROCESS AS GIN CHAR	
2748	6B87	3A	88 FF	LDA ZCHAR ;FETCH CHAR	
2749	6B8A	.	.	;ONLY VALID ESC SEQ IN GIN MODE ARE ESC-FF(PAGE)	
2750	6B8A	.	.	;AND ESC-ENG (READ WITHOUT WAIT)	
2751	6B8A	FE	0C .	CPI 14Q ;FF?	
2752	6B8C	CA	1D 6A	JZ PAGE ;YES, DO A PAGE	
2753	6B8F	FE	05 .	CPI 5Q ;ENG??	
2754	6B91	CA	5B 6B	JZ GINENG ;YES, SEND CURSOR POSITION	
2755	6B94	.	.	;IGNORE ANY OTHER CHAR AFTER ESCAPE	
2756	6B94	21	A0 61	LXI H,GINTAB ;RESTORE GIN TABLE	
2757	6B97	C3	86 63	JMP SETRTB	
2758	6B9A	.	.	;*****	
2759	6B9A	.	.	;GINCR--CARRIAGE RETURN RECEIVED IN GIN MODE	
2760	6B9A	.	.	;TERMINATE GIN IF FROM REMOTE	
2761	6B9A	.	.	;*****	
2762	6B9A	.	.	GINCR EQU \$	
2763	6B9A	CD	C3 00	CALL ZDCIO ;FROM KEYBOARD?	
2764	6B9D	CA	1A 6B	JZ GINCH1 ;YES, USE AS GIN CHAR	
2765	6BA0	.	.	;TERMINATE GIN MODE	
2766	6BA0	CD	A9 6B	CALL CLRSUP ;CLEAR ECHOPLEX SUPRESS	
2767	6BA3	CD	37 6B	CALL GINCH3 ;TERMINATE GIN	
2768	6BA6	C3	C0 00	JMP ZCRRET ;DO THE CARRIAGE RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 79
=====
2770      6BA9      . . .      ;*****
2771      6BA9      . . .      ; CLRSUP--CLEAR CHARACTER (ECHOPLEX) SUPPRESS
2772      6BA9      . . .      ; ENTRY--DONT CARE
2773      6BA9      . . .      ; EXIT A DESTROYED
2774      6BA9      . . .      ;*****
2775      6BA9      . . .      CLRSUP EQU $
2776      6BA9      3A AD 90      LDA TKFLGS
2777      6BAC      E6 DF .      ANI -1-SUPCHR
2778      6BAE      32 AD 90      STA TKFLGS
2779      6BB1      C9 . .      RET
2780      6BB2      . . .      ;*****
2781      6BB2      . . .      ; SNDTEK--SEND TEK COORDINATES AND TERMINATOR
2782      6BB2      . . .      ; ENTRY--HL = X COORD, DE = Y COORD
2783      6BB2      . . .      ;*****
2784      6BB2      . . .      SNDTEK EQU $
2785      6BB2      D5 . .      PUSH D ;SAVE Y
2786      6BB3      CD D0 6B      CALL SNDTKX ;SEND X COORD
2787      6BB6      E1 . .      POP H
2788      6BB7      CD FA 6B      CALL SNDTKY ;SEND Y COORD
2789      6BBA      . . .      ; SEND TERMINATOR
2790      6BBA      3A DC FB      LDA TEKTRM ;FETCH TERMINATOR
2791      6BB0      E6 02 .      ANI SNDNIL ;SEND NOTHING?
2792      6BBF      C0 . .      RNZ ;YES--DONE
2793      6BC0      3E 0D .      MVI A,15Q ;SEND A CARRIAGE RET
2794      6BC2      CD 60 6C      CALL XPUTDC
2795      6BC5      3A DC FB      LDA TEKTRM ;SEND EOT AFTER CR?
2796      6BC8      E6 01 .      ANI SNDEOT
2797      6BCA      C8 . .      RZ ;NO, DONE
2798      6BCB      3E 04 .      MVI A,4Q ;YES, SEND AN EOT
2799      6BCD      C3 60 6C      JMP XPUTDC
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 80
2801	68D0	.	.	;*****	
2802	68D0	.	.	; SNDTKX--SEND TEK X COORDINATE. SCALE IF	
2803	68D0	.	.	; IN SCALED MODE	
2804	68D0	.	.	; ENTRY HL = COORDINATE	
2805	68D0	.	.	;*****	
2806	68D0	.	.	SNDTKX EQU \$	
2807	68D0	CD	5C 69	CALL CKSCLD ;IN SCALED TEK MODE?	
2808	68D3	CA	E0 6B	JZ TKX005 ;NO, PROCESS UNSCALED X	
2809	68D6	.	.	; MULTIPLY X COORDINATE BY 2 TO CONVERT FROM	
2810	68D6	.	.	; 0-511 TO 0-1023	
2811	68D6	11	FF 01	LXI D,511 ;DO BOUNDS CHECK FOR CURSOR	
2812	68D9	CD	41 A3	CALL BNDCHK ;INSURE X LT 512	
2813	68DC	.	.	; MULTIPLY X BY 2	
2814	68DC	29	.	DAD H	
2815	68DD	C3	E5 6B	JMP TKX010	
2816	68E0	.	.	; ADD RELOC ORIGIN TO UNSCALED COORDINATE	
2817	68E0	.	.	TKX005 EQU \$	
2818	68E0	EB	.	XCHG	
2819	68E1	2A	9A 90	LHLD XORG	
2820	68E4	19	.	DAD D ;HL = X + RELOC X ORG	
2821	68E5	.	.	TKX010 EQU \$	
2822	68E5	.	.	; SEND 5 MSBITS	
2823	68E5	E5	.	PUSH H ;SAVE COORDINATE	
2824	68E6	29	.	DAD H ;PUT 5 MSB INTO H	
2825	68E7	29	.	DAD H	
2826	68E8	29	.	DAD H	
2827	68E9	3E	1F .	MVI A,37Q ;WANT 5 MSB ONLY	
2828	68E8	A4	.	ANA H	
2829	68EC	F6	20 .	ORI 40Q ;ADD TAG BITS	
2830	68EE	CD	60 6C	CALL XPUTDC ;SEND 5 MSB	
2831	68F1	.	.	; SEND 5 LSB	
2832	68F1	E1	.	POP H ;RESTORE COORD	
2833	68F2	3E	1F .	MVI A,37Q ;WANT 5 LSB ONLY	
2834	68F4	A5	.	ANA L	
2835	68F5	F6	20 .	ORI 40Q ;ADD TAG BITS	
2836	68F7	C3	60 6C	JMP XPUTDC ;SEND 5 LSB	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 81
=====
2838      6BFA      . . .      ;*****
2839      6BFA      . . .      ; SNDTKY--SEND TEK Y COORD, SCALED IF NECESSARY
2840      6BFA      . . .      ; ENTRY HL = Y COORD
2841      6BFA      . . .      ;*****
2842      6BFA      . . .      SNDTKY EQU $
2843      6BFA      CD 5C 69      CALL CKSCLO      ;IN SCALED TEK MODE?
2844      6BFD      C2 08 6C      JNZ TKY010      ;YES,SCALE Y COORD
2845      6C00      . . .      ; ADD RELOCATABLE ORIGIN TO UNSCALED COORD
2846      6C00      EB . .      XCHG
2847      6C01      2A 98 90      LHLD YORG
2848      6C04      19 . .      DAD D
2849      6C05      C3 E5 6B      JMP TKX010      ;SEND Y COORD
2850      6C08      . . .      TKY010 EQU $
2851      6C08      . . .      ; SCALE Y COORD--MULTIPLY BY 128/59
2852      6C08      . . .      ; FIRST MULTIPLY BY 128 (SHIFT LEFT 7)
2853      6C08      29 . .      DAD H
2854      6C09      29 . .      DAD H
2855      6C0A      29 . .      DAD H
2856      6C0B      29 . .      DAD H
2857      6C0C      29 . .      DAD H
2858      6C0D      29 . .      DAD H
2859      6C0E      29 . .      DAD H
2860      6C0F      . . .      ; DIVIDE BY 59 BY REPEATED SUBTRACTION
2861      6C0F      . . .      ; INITALLY SUBTRACT UNTIL SIGN BIT IS +
2862      6C0F      . . .      ; (MAY BE - CAUSE OF MULTIPLICATION BY 128)
2863      6C0F      01 C5 FF      LXI B,-59
2864      6C12      11 00 00      LXI D,0      ;D = RESULT
2865      6C15      7C . .      MOV A,H      ;SIGN BIT SET
2866      6C16      B7 . .      ORA A
2867      6C17      F2 21 6C      JP TKY020      ;NO, DONT DO INITAL SUB.
2868      6C1A      . . .      ; SUBTRACT UNTIL SIGN BIT GOES +
2869      6C1A      . . .      TKY015 EQU $
2870      6C1A      09 . .      DAD B
2871      6C1B      13 . .      INX D      ;BUMP RESULT
2872      6C1C      7C . .      MOV A,H      ;SIGN BIT STILL -?
2873      6C1D      B7 . .      ORA A
2874      6C1E      FA 1A 6C      JM TKY015      ;YES, CONTINUE SUBTRACTION
2875      6C21      . . .      ; SUBTRACT UNTIL SIGN GOES -
2876      6C21      . . .      TKY020 EQU $
2877      6C21      09 . .      DAD B      ;SUBTRACT 59
2878      6C22      13 . .      INX D      ;BUMP RESULT
2879      6C23      7C . .      MOV A,H      ;SIGN BIT - YET?
2880      6C24      B7 . .      ORA A
2881      6C25      F2 21 6C      JP TKY020      ;NO, CONTINUE SUBTRACTION
2882      6C28      . . .      ; Y COORD = Y * 128/59
2883      6C28      1B . .      DCX D      ;SUBTRACTED ONCE TOO MANY
2884      6C29      EB . .      XCHG      ;HL = NEW Y COORD
2885      6C2A      C3 E5 6B      JMP TKX010      ;SEND COORD
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  82
=====
2887      6C2D      . . .      ;*****
2888      6C2D      . . .      ; TKHC--ESC ETB RECEIVED, SEND GRAPHICS IMAGE TO
2889      6C2D      . . .      ; HARDCOPY. MUST HAVE EITHER VIDEO PRINTER OR
2890      6C2D      . . .      ; 8 BIT DUPLEX CARD FOR BOISE PRINTER.
2891      6C2D      . . .      ; IF ALT I/O ROM IS PRESENT, THE BOISE PRINTER
2892      6C2D      . . .      ; IS ASSUMED.
2893      6C2D      . . .      ; ENTRY--DONT CARE
2894      6C2D      . . .      ; EXIT---ALL REGISTERS DESTROYED
2895      6C2D      . . .      ;*****
2896      6C2D      . . .      TKHC EQU $
2897      6C2D      CD 4F 00      CALL ZESCND ;RESET RANGE TABLES
2898      6C30      CD 56 69      CALL CHKTEK ;IN TEK MODE?
2899      6C33      C8 . .      RZ ;NO, EXIT
2900      6C34      CD A9 6A      CALL CHKPTR ;ANY GRAPHICS PRINTER THERE?
2901      6C37      C8 . .      RZ ;NO
2902      6C38      . . .      ; ASSUME ALTERNATE I/O FOR BOISE PRINTER THERE
2903      6C38      21 11 92      LXI H,ZPUTAL ;'PUT' ROUTINE ADDRESS
2904      6C38      CD A5 00      CALL ZIORGO ;TRY TO EXECUTE
2905      6C3E      D0 . .      RNC ;EXIT IF ROUTINE WAS EXECUTE
2906      6C3F      . . .      ; ALT I/O, BOISE PRINTER NOT PRESENT
2907      6C3F      . . .      ; INITIATE PRINT OPERATION ON VIDEO PRINTER
2908      6C3F      AF . .      XRA A ;SEND SINGLE CHAR
2909      6C40      32 20 8D      STA PTR0T1 ;TO START THE PRINT
2910      6C43      C9 . .      RET
2911      6C44      . . .      ;*****
2912      6C44      . . .      ; DIV128--DIVIDE HL BY 128 BY SHIFTING RIGHT
2913      6C44      . . .      ; ONLY CALLED BY 'LOWX' ROUTINE
2914      6C44      . . .      ; ENTRY C = 7 FOR 7 SHIFTS
2915      6C44      . . .      ; EXIT HL = HL /128
2916      6C44      . . .      ;*****
2917      6C44      . . .      DIV128 EQU $
2918      6C44      0D . .      DCR C ;ALL 7 DONE?
2919      6C45      F8 . .      RM ;YES, EXIT
2920      6C46      7C . .      MOV A,H ;SHIFT MSBYTE
2921      6C47      1F . .      RAR
2922      6C48      67 . .      MOV H,A
2923      6C49      7D . .      MOV A,L ;SHIFT LSBYTE
2924      6C4A      1F . .      RAR
2925      6C4B      6F . .      MOV L,A
2926      6C4C      C3 44 6C      JMP DIV128 ;GO THROUGH AGAIN
2927      6C4F      . . .      ; INSERT NOPS TO MAINTAIN ADDRESSES FOR GR68A
2928      6C4F      . . .      ; ROMS
2929      6C4F      00 00 00      DB 0,0,0,0,0,0,0,0
2930      6C57      . . .      ;*****
2931      6C57      . . .      ; TEKHC--ESC ETB RECEIVED IN GRAPHICS MODE
2932      6C57      . . .      ;*****
2933      6C57      . . .      TEKHC EQU $
2934      6C57      CD 2D 6C      CALL TKHC ;MAKE THE HARD COPY
2935      6C5A      21 6C 61      LXI H,TKGSTB ;RESTORE GS TABLE
2936      6C5D      C3 86 63      JMP SETRTB
=====

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 83
=====
2938     6C60      . . .      ;*****
2939     6C60      . . .      ; XPUTDC--SEND CHAR TO DATACOM AFTER INITIAL
2940     6C60      . . .      ; WAIT (FOR SLOW SYSTEMS)
2941     6C60      . . .      ; ENTRY  A = CHAR
2942     6C60      . . .      ;*****
2943     6C60      . . .      XPUTDC EQU $
2944     6C60      F5 . .      PUSH PSW          ;SAVE THE CHAR
2945     6C61      2E 01 .      MVI L,1          ;WAIT 10 MS
2946     6C63      CD B4 00      CALL ZDELAY      ;BEFORE SENDING
2947     6C66      F1 . .      POP PSW          ;RECALL THE CHAR
2948     6C67      C3 7C 00      JMP ZPUTDC       ;SEND THE CHAR
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2950	6C6A	.	.	;*****	84
2951	6C6A	.	.	; LBLSEQ--ESC * L RECEIVED, SET UP FOR LABEL	
2952	6C6A	.	.	;*****	
2953	6C6A	.	.	LBLSEQ EQU \$	
2954	6C6A	21	BC 61	LXI H,LBLTAB ;NEW RANGE TABLE	
2955	6C6D	22	D2 FF	SHLD ZRNGTA	
2956	6C70	AF	.	XRA A ;ZERO THE LABEL COUNT	
2957	6C71	32	74 90	STA LBLCTR	
2958	6C74	3E	80 .	MVI A,LABEL ;SET LABEL IN PROGRESS FLAG	
2959	6C76	C3	5A A2	JMP STFLG6	
2960	6C79	.	.	;*****	
2961	6C79	.	.	; LBLCR--CR RECEIVED IN LABEL ESC SEQ	
2962	6C79	.	.	; EMPTY BUFFER, DO THE RETURN, WAIT FOR LFF	
2963	6C79	.	.	;*****	
2964	6C79	.	.	LBLCR EQU \$	
2965	6C79	CD	AF 9C	CALL XCR ;DO THE RETURN	
2966	6C7C	.	.	LBL1 EQU \$	
2967	6C7C	21	D0 61	LXI H,LBLTB2 ;EXAMINE NEXT CHAR FOR CR OR	
2968	6C7F	F2	86 63	JP SETRTB	
2969	6C82	.	.	;*****	
2970	6C82	.	.	; LBLLF--SAME AS LBLCR	
2971	6C82	.	.	;*****	
2972	6C82	.	.	LBLLF EQU \$	
2973	6C82	CD	F9 9C	CALL XLF ;DO THE LINE FEED	
2974	6C85	C3	7C 6C	JMP LBL1 ;EXAMINE NEXT CHAR FOR CR	
2975	6C88	.	.	;*****	
2976	6C88	.	.	; LBLCR2,LBLLF2--DO THE CR OR LF, THEN EXIT	
2977	6C88	.	.	; LABEL MODE	
2978	6C88	.	.	;*****	
2979	6C88	.	.	LBLCR2 EQU \$	
2980	6C88	CD	AF 9C	CALL XCR	
2981	6C88	.	.	LBL2 EQU \$	
2982	6C88	CD	9D 6C	CALL LBLOFF	
2983	6C8E	C3	4F 00	JMP ZESCND	
2984	6C91	.	.	LBLLF2 EQU \$	
2985	6C91	CD	F9 9C	CALL XLF	
2986	6C94	C3	8B 6C	JMP LBL2	
2987	6C97	.	.	;*****	
2988	6C97	.	.	; LBLESC--ESCAPE RECEIVED IN ESC*L SEQ	
2989	6C97	.	.	; TURN LABEL MODE OFF BEFORE DOING ESC	
2990	6C97	.	.	;*****	
2991	6C97	.	.	LBLESC EQU \$	
2992	6C97	CD	9D 6C	CALL LBLOFF	
2993	6C9A	C3	87 00	JMP ZESCAP ;PROCESS THE ESCAPE	
2994	6C9D	.	.	;*****	
2995	6C9D	.	.	; LBLOFF--ABORT PENDING LABEL, IF ANY	
2996	6C9D	.	.	;*****	
2997	6C9D	.	.	LBLOFF EQU \$	
2998	6C9D	AF	.	XRA A ;CLEAR CHARACTER COUNT	
2999	6C9E	32	74 90	STA LBLCTR	

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS          PAGE 85
=====
3000      6CA1      3E 80      .          MVI A,LABEL
3001      6CA3      C3 60      A2         JMP CLFLG6
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 86
=====
3003      6CA6      . . .      ;*****
3004      6CA6      . . .      ; GPARAM--PROCESS ASCII PARAMETER IN GRAPHICS
3005      6CA6      . . .      ; ESCAPE SEQUENCE
3006      6CA6      . . .      ; ENTRY ZCHAR = CHARACTER
3007      6CA6      . . .      ;*****
3008      6CA6      . . .      GPARAM EQU $
3009      6CA6      3A 88 FF      LDA ZCHAR      ;FETCH PARAMETER
3010      6CA9      FE 30 .      CPI 60Q        ;DIGIT?
3011      6CAB      DA B3 6C      JC GPM010      ;NO, TOO SMALL
3012      6CAE      FE 3A .      CPI 72Q        ;DIGIT?
3013      6CB0      DA C5 6C      JC GPM020      ;YES, PROCESS
3014      6CB3      . . .      GPM010 EQU $
3015      6CB3      FE 2D .      CPI 55Q        ;- SIGN?
3016      6CB5      CA E3 6C      JZ GPM030      ;YES
3017      6CB8      FE 2B .      CPI 53Q        ;+ SIGN?
3018      6CBA      CA E3 6C      JZ GPM030      ;YES
3019      6CBD      FE 2E .      CPI 56Q        ;DECIMAL POINT?
3020      6CBF      CA F9 6C      JZ GPM040      ;YES
3021      6CC2      . . .      ; CHAR IS NOT PART OF A NUMBER, SO TERMINATE
3022      6CC2      . . .      ; NUMBER BEING BUILT (IF THERE IS ONE)
3023      6CC2      C3 15 6D      JMP STOPPM
3024      6CC5      . . .      ; CHARACTER IS A DIGIT
3025      6CC5      . . .      GPM020 EQU $
3026      6CC5      4F . .      MOV C,A        ;SAVE DIGIT
3027      6CC6      3A 96 90      LDA GFLGS7     ;IS A NUMBER IN PROGRESS?
3028      6CC9      E6 01 .      ANI NIP
3029      6CCB      CC 06 6D      CZ BGNPRM      ;IF NOT, START ONE
3030      6CCE      3E 10 .      MVI A,HAVED    ;SET HAVE DIGIT FLAG
3031      6CD0      CD 67 A2      CALL STFLG7
3032      6CD3      E6 08 .      ANI HAVEP      ;HAS DEC POINT BEEN FOUND?
3033      6CD5      C0 . .      RNZ            ;YES, IGNORE DIGIT
3034      6CD6      . . .      ; CONVERT TO BINARY
3035      6CD6      79 . .      MOV A,C        ;RECALL CHAR
3036      6CD7      2A B7 90      LHLD TMPBUF    ;TEMPORARY ACCUMULATOR
3037      6CDA      EB . .      XCHG           ;DE = ACCUMUALTOR
3038      6CDB      CD 4E AF      CALL BCDBIN    ;CONVERT TO BINARY
3039      6CDE      EB . .      XCHG           ;HL = VALUE
3040      6CDF      22 B7 90      SHLD TMPBUF    ;STORE NEW ACCUMULATED VALUE
3041      6CE2      C9 . .      RET
3042      6CE3      . . .      ; PROCESS + OR - SIGN
3043      6CE3      . . .      GPM030 EQU $
3044      6CE3      3A 96 90      LDA GFLGS7     ;NUMBER CURRENTLY IN
3045      6CE6      E6 01 .      ANI NIP        ;PROGRESS?
3046      6CE8      C4 15 6D      CNZ STOPPM     ;YES, TERMINATE IT
3047      6CEB      CD 06 6D      CALL BGNPRM    ;START A NEW PARAMETER
3048      6CEE      . . .      ; IF SIGN IS -, SET FLAG
3049      6CEE      3A 88 FF      LDA ZCHAR      ;WAS IT -?
3050      6CF1      FE 2D .      CPI 55Q
3051      6CF3      C0 . .      RNZ            ;NO, DONT SET FLAG
3052      6CF4      3E 20 .      MVI A,MINUS   ;YES

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 87
3053	6CF6	C3	67	A2	JMP STFLG7
3054	6CF9	.	.	.	; PROCESS DECIMAL POINT
3055	6CF9	.	.	.	GPM040 EQU \$
3056	6CF9	3A	96	90	LDA GFLGS7 ;NUMBER IN PROGRESS?
3057	6CFC	E6	01	.	ANI NIP
3058	6CFE	CC	06	6D	CZ BGNPRM ;START ONE IF NOT
3059	6D01	.	.	.	; SET FLAG TO IGNORE ALL DIGITS AFTER DEC POINT
3060	6D01	3E	08	.	MVI A,HAVEP ;SET HAVE '.' FLAG
3061	6D03	C3	67	A2	JMP STFLG7
3062	6D06	.	.	.	;*****
3063	6D06	.	.	.	; BGNPRM--BEGIN BUILDING AN ASCII PARAMETER
3064	6D06	.	.	.	;*****
3065	6D06	.	.	.	BGNPRM EQU \$
3066	6D06	21	00	00	LXI H,0 ;CLEAR TEMPORARY ACCUMULATOR
3067	6D09	22	87	90	SHLD TMPBUF
3068	6D0C	3E	38	.	MVI A,HAVED+HAVEP+MINUS ;CLEAR FLAGS
3069	6D0E	CD	6D	A2	CALL CLFLG7
3070	6D11	F6	01	.	ORI NIP ;SET NUMBER IN PROGRESS
3071	6D13	77	.	.	MOV M,A
3072	6D14	C9	.	.	RET
3073	6D15	.	.	.	;*****
3074	6D15	.	.	.	; STOPPM--STOP BUILDING PARAMETER,
3075	6D15	.	.	.	; PUT VALUE INTO PARAMETER BUFFER
3076	6D15	.	.	.	; UPDATE PARAMETER COUNT
3077	6D15	.	.	.	;*****
3078	6D15	.	.	.	STOPPM EQU \$
3079	6D15	3A	96	90	LDA GFLGS7 ;WERE ANY DIGITS RECEIVED?
3080	6D18	4F	.	.	MOV C,A ;(SAVE FLAGS IN C)
3081	6D19	E6	10	.	ANI HAVED
3082	6D1B	CA	3E	6D	JZ SPM010 ;NO, IGNORE PARAMETER
3083	6D1E	.	.	.	; UPDATE PARAMETER INDEX
3084	6D1E	21	86	90	LXI H,PRMDEX
3085	6D21	7E	.	.	MOV A,M ;FETCH PARAMETER COUNT
3086	6D22	FE	08	.	CPI MAXPRM ;HAVE MAX NUMBER ALREADY?
3087	6D24	D2	3E	6D	JNC SPM010 ;YES, IGNORE PARAMETER
3088	6D27	34	.	.	INR M ;UPDATE PARAMETER COUNT
3089	6D28	.	.	.	; USING PARAMETER COUNT AS INDEX, STORE
3090	6D28	.	.	.	; PARAMETER VALUE IN PARAMETER BUFFER
3091	6D28	87	.	.	ADD A ;(2 BYTES/ENTRY)
3092	6D29	5F	.	.	MOV E,A
3093	6D2A	16	00	.	MVI D,0 ;DE = INDEX
3094	6D2C	21	89	90	LXI H,PRMBUF ;BASE OF BUFFER
3095	6D2F	19	.	.	DAD D
3096	6D30	EB	.	.	XCHG ;DE = POINTER TO BUFFER SLOT
3097	6D31	2A	87	90	LHLD TMPBUF ;FETCH PARAMETER VALUE
3098	6D34	3E	20	.	MVI A,MINUS ;WAS A - SIGN RECEIVED?
3099	6D36	A1	.	.	ANA C
3100	6D37	C4	09	A3	CNZ NEGATE ;YES, NEGATE PARAMETER VALUE
3101	6D3A	EB	.	.	XCHG ;DE = VALUE, HL = POINTER
3102	6D3B	73	.	.	MOV M,E ;STORE LSBYTE

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE	88
3103	6D3C	23	.	INX H		
3104	6D3D	72	.	MOV M,D ;STORE MSYTE		
3105	6D3E	.	.	; CLEAR PARAMETER FLAGS FOR NEXT VALUE		
3106	6D3E	.	.	SPM010 EQU \$		
3107	6D3E	3E	39	MVI A,NIP+HAVED+HAVEP+MINUS		
3108	6D40	C3	6D	A2 JMP CLFLG7		
3109	6D43	.	.	;*****		
3110	6D43	.	.	; GETPRM--GET SINGLE PARAMETER		
3111	6D43	.	.	; DO CHECK TO INSURE ONLY ONE PARAM RECEIVED		
3112	6D43	.	.	; DO CHECK ON MAXIMUM VALUE		
3113	6D43	.	.	; ENTRY A = MAX VALUE (+ ONLY)		
3114	6D43	.	.	; EXIT A = PARAMETER		
3115	6D43	.	.	; NZ => WRONG NUMBER OF PARAMS OR > MAX		
3116	6D43	.	.	; Z => OK		
3117	6D43	.	.	;*****		
3118	6D43	.	.	GETPRM EQU \$		
3119	6D43	F5	.	PUSH PSW ;SAVE MAX VALUE		
3120	6D44	CD	15	6D CALL STOPPM ;TERMINATE PARAM IN PROGRESS		
3121	6D47	F1	.	POP PSW ;RECALL VALUE		
3122	6D48	3C	.	INR A ;MAX VALUE		
3123	6D49	4F	.	MOV C,A ;LEAVE MAX IN C		
3124	6D4A	3A	B6	90 LDA PRMDEX ;RECEIVED ONE PARAMETER ONLY		
3125	6D4D	3D	.	DCR A		
3126	6D4E	C0	.	RNZ ;NO		
3127	6D4F	3A	BA	90 LDA PRMBUF+1 ;IS MSBYTE OF PARAM = 0?		
3128	6D52	B7	.	ORA A		
3129	6D53	C0	.	RNZ ;NO		
3130	6D54	3A	B9	90 LDA PRMBUF ;FETCH PARAMETER VALUE		
3131	6D57	B9	.	CMP C ;IS IT LT MAX?		
3132	6D58	D2	5F	6D JNC GPR010 ;NO, RETURN BAD		
3133	6D5B	.	.	; PARAMETER IS OK, SET Z FLAG		
3134	6D5B	4F	.	MOV C,A ;SAVE VALUE		
3135	6D5C	AF	.	XRA A ;SET Z		
3136	6D5D	79	.	MOV A,C ;RESTORE		
3137	6D5E	C9	.	RET		
3138	6D5F	.	.	GPR010 EQU \$		
3139	6D5F	F6	FF	ORI 377Q ;SET NZ		
3140	6D61	C9	.	RET		
3141	6D62	.	.	;*****		
3142	6D62	.	.	; PRMCNT--SEE IF PROPER NUMBER OF PARAMETERS		
3143	6D62	.	.	; HAVE BEEN RECEIVED		
3144	6D62	.	.	; ENTRY A = NUMBER OF PARAMETERS		
3145	6D62	.	.	; EXIT NZ => WRONG NUMBER RECEIVED		
3146	6D62	.	.	;*****		
3147	6D62	.	.	PRMCNT EQU \$		
3148	6D62	F5	.	PUSH PSW ;SAVE MAX		
3149	6D63	CD	15	6D CALL STOPPM ;STOP PARAMETER IN PROGRESS		
3150	6D66	F1	.	POP PSW		
3151	6D67	21	B6	90 LXI H,PRMDEX ;PTR TO NUMBER OF PARAMS		
3152	6D6A	BE	.	CMP M		

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
3155	6D6C	.	.	;*****	90
3156	6D6C	.	.	; DSPSEQ--ESC * D RECEIVED, SET UP FOR DISPLAY	
3157	6D6C	.	.	; CONTROL ESCAPE SEQUENCE	
3158	6D6C	.	.	;*****	
3159	6D6C	.	.	DSPSEQ EQU \$	
3160	6D6C	21	D2 60	LXI H,DSPTAB ;USE DISPLAY RANGE TABLE	
3161	6D6F	C3	86 63	JMP SETRTB	
3162	6D72	.	.	;	
3163	6D72	.	.	;*****	
3164	6D72	.	.	; GCLEAR--CLEAR THE GRAPHIC IMAGE MEMORY	
3165	6D72	.	.	;*****	
3166	6D72	.	.	GCLEAR EQU \$	
3167	6D72	CD	78 6D	CALL GCLR1	
3168	6D75	C3	C1 99	JMP GEXIT	
3169	6D78	.	.	GCLR1 EQU \$;(INTERNAL ENTRY)	
3170	6D78	3E	09 .	MVI A,CLRMEM ;HCEJK BITS TO CLEAR SCREEN	
3171	6D7A	C3	85 6D	JMP SET1	
3172	6D7D	.	.	;*****	
3173	6D7D	.	.	; GSET--SET THE GRAPHICS IMAGE MEMORY	
3174	6D7D	.	.	;*****	
3175	6D7D	.	.	GSET EQU \$	
3176	6D7D	CD	83 6D	CALL GSET1	
3177	6D80	C3	C1 99	JMP GEXIT	
3178	6D83	.	.	GSET1 EQU \$;(INTERNAL ENTRY)	
3179	6D83	3E	0A .	MVI A,SETMEM ;HCEJK TO SET MEMORY	
3180	6D85	.	.	SET1 EQU \$	
3181	6D85	F5	.	PUSH PSW ;SAVE HCEJK BITS	
3182	6D86	3E	02 .	MVI A,SUPR1 ;SET SUPPRESS BIT	
3183	6D88	CD	8A 9E	CALL SUPRGC ;SUPPRESS THE CURSOR	
3184	6D8B	F1	.	POP PSW ;RECALL HCEJK BITS	
3185	6D8C	F5	.	PUSH PSW ;SAVE HCEJK BITS USED	
3186	6D8D	32	41 89	STA HCEJK ;SEND CONTROL BITS TO HW	
3187	6D90	2E	02 .	MVI L,2 ;SET DELAY OF 20 MS	
3188	6D92	CD	B4 00	CALL ZDELAY ;WAIT 20MS TO DO ENTIRE MEM	
3189	6D95	F1	.	POP PSW ;RECALL HCEJK BITS	
3190	6D96	E6	F7 .	ANI 3670 ;DELETE C BIT	
3191	6D98	32	41 89	STA HCEJK ;SEND TO HW TO STOP CLR/SET	
3192	6D9B	3A	B5 90	LDA CURMOD ;RESTORE PREVIOUS MODE	
3193	6D9E	32	41 89	STA HCEJK	
3194	6DA1	3E	02 .	MVI A,SUPR1 ;UNSUPPRESS THE GRAPHICS	
3195	6DA3	C3	B5 9E	JMP ENABGC ;CURSOR	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
3197	6DA6	.	.	;*****
3198	6DA6	.	.	; GVON--TURN GRAPHICS VIDEO ON
3199	6DA6	.	.	;*****
3200	6DA6	.	.	GVON EQU \$
3201	6DA6	CD	AC 6D	CALL GVON1
3202	6DA9	C3	C1 99	JMP GEXIT
3203	6DAC	.	.	GVON1 EQU \$;(INTERNAL ENTRY)
3204	6DAC	3E	04 .	MVI A,SUPRZM ;UNSUPRESS ZOOM
3205	6DAE	CD	46 A2	CALL CLFLG5
3206	6DB1	CD	87 A2	CALL WAIT ;INSURE HW IDLE
3207	6DB4	.	.	;IF ZOOM WAS ON, IT MUST BE RESTARTED BEFORE
3208	6DB4	.	.	;TURNING DISPLAY BACK ON
3209	6DB4	E6	02 .	ANI WANTZM ;WAS ZOOM ON?
3210	6DB6	CA	C4 6D	JZ GV005 ;NO,DONT TRY TO RESTART IT
3211	6DB9	.	.	;TURN ZOOM ON, AND WAIT A FRAME TO INSURE ITS ON
3212	6DB9	3E	01 .	MVI A,GCM1 ;FAKE A CURSOR MOVE TO INSUR
3213	6DBB	CD	40 A2	CALL STFLG5 ;ZOOM GOES ON
3214	6DBE	CD	CA A0	CALL ZMUPDA ;DO ZOOM UPDATE
3215	6DC1	CD	74 A2	CALL SNOGCF ;SEND ZOOM, CURSOR FLAGS
3216	6DC4	.	.	GV005 EQU \$
3217	6DC4	CD	92 A2	CALL VRWAIT ;WAIT FOR END OF FRAME
3218	6DC7	3E	10 .	MVI A,GVENAB ;VIDEO ENABLE BIT
3219	6DC9	21	B5 90	LXI H,CURMOD ;SET H BIT IN HCEJK
3220	6DCC	B6	. .	ORA M ;FLAGS
3221	6DCD	77	. .	MOV M,A
3222	6DCE	CD	32 72	CALL SNDMOD ;SEND MODE TO HW
3223	6DD1	.	.	;CHANGE DOTS/SCAN LINE TO 4 IF NO ZOOM, 3 IF ZOOM
3224	6DD1	21	FC FF	LXI H,NRMVEC ;ASSUME NOT IN ZOOM
3225	6DD4	3A	B1 90	LDA GFLGS2 ;LOAD HW STATUS
3226	6DD7	E6	02 .	ANI ZOOM ;IN ZOOM MODE
3227	6DD9	CA	DF 6D	JZ GV010 ;NO
3228	6DDC	21	FD FF	LXI H,SLOVEC ;YES, SET FOR FEWER DOTS
3229	6DDF	.	.	GV010 EQU \$
3230	6DDF	22	02 89	SHLD VDC ;SEND NEW DOTS/LINE TO HW
3231	6DE2	3E	04 .	MVI A,SUPR2 ;UNSUPRESS THE GRAPHICS
3232	6DE4	C3	B5 9E	JMP ENABGC ;CURSOR

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 92
3234	6DE7	. . .	;*****	
3235	6DE7	. . .	; GVOFF--TURN GRAPHICS VIDEO OFF	
3236	6DE7	. . .	;*****	
3237	6DE7	. . .	GVOFF EQU \$	
3238	6DE7	CD ED 6D	CALL GVOFF1	
3239	6DEA	C3 C1 99	JMP GEXIT	
3240	6DED	. . .	GVOFF1 EQU \$	
3241	6DED	CD 92 A2	CALL VRWAIT ;WAIT FOR END OF FRAME	
3242	6DF0	3E EF .	MVI A,-1-GVENAB ;CLEAR VIDEO ENABLE	
3243	6DF2	21 B5 90	LXI H,CURMOD ;BIT	
3244	6DF5	A6 . .	ANA M	
3245	6DF6	77 . .	MOV M,A	
3246	6DF7	CD 32 72	CALL SNDMOD ;SEND TO HW	
3247	6DFA	. . .	;MUST SUPPRESS ZOOM WHILE VIDEO OFF TO	
3248	6DFA	. . .	;INSURE REFRESH	
3249	6DFA	3E 04 .	MVI A,SUPRZM ;SET ZOOM SUPPRESS BIT	
3250	6DFC	CD 40 A2	CALL STFLG5	
3251	6DFF	E6 02 .	ANI WANTZM ;WAS ZOOM ON?	
3252	6E01	C4 78 6E	CNZ UNZOOM ;IF YES, STOP ZOOMING	
3253	6E04	CD 87 A2	CALL WAIT ;INSURE HW IDLE	
3254	6E07	. . .	;CHANGE #DOTS/SCAN LINE TO 250 FOR HIGH SPEED	
3255	6E07	21 06 FF	LXI H,FSTVEC	
3256	6E0A	22 02 89	SHLD VDC ;SEND NEW DOTS/LINE TO HW	
3257	6E0D	3E 04 .	MVI A,SUPR2 ;SUPPRESS THE GRAPHICS	
3258	6E0F	C3 8A 9E	JMP SUPRGC ;CURSOR	
3259	6E12	. . .	;*****	
3260	6E12	. . .	; ANVON--TURN A/N VIDEO ON	
3261	6E12	. . .	;*****	
3262	6E12	. . .	ANVON EQU \$	
3263	6E12	CD 18 6E	CALL ANVON1	
3264	6E15	C3 C1 99	JMP GEXIT	
3265	6E18	. . .	ANVON1 EQU \$;(INTERNAL ENTRY)	
3266	6E18	3E 20 .	MVI A,AVINHB ;INHIBIT BIT	
3267	6E1A	CD 2C A2	CALL CLFLG1 ;CLEAR IT	
3268	6E1D	CD 19 70	CALL ACON1 ;TURN A/N CURSOR ON TOO	
3269	6E20	C3 43 00	JMP ZRSTDP ;RESTORE NORMAL DISPLAY	
3270	6E23	. . .	;*****	
3271	6E23	. . .	; AVOFF--TURN ALPHA-NUMERIC VIDEO OFF	
3272	6E23	. . .	;*****	
3273	6E23	. . .	ANVOFF EQU \$	
3274	6E23	CD 29 6E	CALL ANVOF1	
3275	6E26	C3 C1 99	JMP GEXIT	
3276	6E29	. . .	ANVOF1 EQU \$	
3277	6E29	21 FF FF	LXI H,DISPST+1 ;PUT NULL MESSAGE	
3278	6E2C	36 CE .	MVI M,ZEOP ;INTO DISPLAY BUFFER	
3279	6E2E	CD 92 A2	CALL VRWAIT ;SYNCH WITH FRAME	
3280	6E31	3E 18 .	MVI A,ZMXROW+1 ;PUT CURSOR OFF SCREEN	
3281	6E33	32 20 87	STA ZIOCRW	
3282	6E36	3E 20 .	MVI A,AVINHB ;SET INHIBIT BIT	
3283	6E38	C3 26 A2	JMP STFLG1	


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 93
=====
3285      6E3B      . . .      ;*****
3286      6E3B      . . .      ; ZON -- TURN ZOOM ON
3287      6E3B      . . .      ;*****
3288      6E3B      . . .      ZON      EQU      $
3289      6E3B      CD 41 6E      CALL ZON1
3290      6E3E      C3 C1 99      JMP GEXIT
3291      6E41      . . .      ZON1     EQU      $      ;(INTERNAL ENTRY)
3292      6E41      3E 82 .      MVI A,WANTZM+NWZOOM ;SET USER WANTS ZOOM
3293      6E43      CD 40 A2      CALL STFLG5 ;AND NEW ZOOM FLAGS
3294      6E46      3A B5 90      LDA CURMOD  ;VIDEO OFF?
3295      6E49      E6 10 .      ANI GVENAB
3296      6E4B      C8 . .      RZ      ;YES, DONT CHANGE SPEED
3297      6E4C      CD 87 A2      CALL WAIT  ;WAIT FOR IDLE HW
3298      6E4F      21 FD FF      LXI H,SLOVEC ;SET FOR FEWER DOTS
3299      6E52      22 02 89      SHLD VDC   ;WHEN IN ZOOM MODE
3300      6E55      C3 7B 6F      JMP EOFRM  ;WAIT FOR END OF FRAME
3301      6E58      . . .      ;*****
3302      6E58      . . .      ; ZOFF -- TURN ZOOM OFF
3303      6E58      . . .      ;*****
3304      6E58      . . .      ZOFF     EQU      $
3305      6E58      CD 5E 6E      CALL ZOFF1
3306      6E5B      C3 C1 99      JMP GEXIT
3307      6E5E      . . .      ZOFF1    EQU      $      ;(INTERNAL ENTRY)
3308      6E5E      3E 02 .      MVI A,WANTZM
3309      6E60      CD 46 A2      CALL CLFLG5 ;CLEAR WANT ZOOM FLAG
3310      6E63      CD 78 6E      CALL UNZOOM ;STOP ZOOMING
3311      6E66      3A B5 90      LDA CURMOD  ;VIDEO OFF?
3312      6E69      E6 10 .      ANI GVENAB
3313      6E6B      C8 . .      RZ      ;YES, DONT CHANGE SPEED
3314      6E6C      CD 87 A2      CALL WAIT  ;INSURE HW IDLE
3315      6E6F      21 FC FF      LXI H,NRMVEC ;SET TO NORMAL NUMBER OF DOT
3316      6E72      22 02 89      SHLD VDC
3317      6E75      C3 7B 6F      JMP EOFRM  ;WAIT FOR END OF FRAME
3318      6E78      . . .      ;*****
3319      6E78      . . .      ; UNZOOM -- STOP ZOOMING
3320      6E78      . . .      ;*****
3321      6E78      . . .      UNZOOM   EQU      $
3322      6E78      CD 87 A2      CALL WAIT  ;INSURE HW IDLE
3323      6E7B      3A B1 90      LDA GFLG52 ;CLEAR ZOOM BIT
3324      6E7E      E6 FD .      ANI 377Q-ZOOM
3325      6E80      32 B1 90      STA GFLG52
3326      6E83      32 20 89      STA HWFLG5 ;SEND TO HW
3327      6E86      C3 92 A2      JMP VRWAIT ;WAIT FOR END OF FRAME
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
3329	6E89	.	.	;*****	94
3330	6E89	.	.	; ZSIZE -- SET ZOOM SIZE, AND COMPUTE NEW	
3331	6E89	.	.	; PARAMETERS FOR NEW SIZE	
3332	6E89	.	.	;*****	
3333	6E89	.	.	ZSIZE EQU \$	
3334	6E89	3E	10	MVI A,16 ;MAX SIZE	
3335	6E88	CD	43 6D	CALL GETPRM ;GET VALUE	
3336	6E8E	C2	C1 99	JNZ GEXIT ;IGNORE IF BAD	
3337	6E91	3D	.	DCR A ;WANT 0-15, NOT 1-16	
3338	6E92	FA	C1 99	JM GEXIT ;IGNORE IF VALUE 0	
3339	6E95	CD	9E 6E	CALL NWSIZE ;COMPUTE NEW PARAMETERS	
3340	6E98	CD	7B 6F	CALL EOFRM ;WAIT FOR END OF FRAME	
3341	6E98	C3	C1 99	JMP GEXIT	
3342	6E9E	.	.	;*****	
3343	6E9E	.	.	; NWSIZE -- COMPUTE NEW ZOOM PARAMETERS FROM	
3344	6E9E	.	.	; NEW ZOOM MAGNIFICATION	
3345	6E9E	.	.	; ENTRY A = NEW SIZE (0-15 ONLY)	
3346	6E9E	.	.	;*****	
3347	6E9E	.	.	NWSIZE EQU \$	
3348	6E9E	32	E1 FB	STA MAG ;STORE NEW MAGNIFICATION	
3349	6EA1	B7	.	ORA A ;IS SIZE = 1X ?	
3350	6EA2	CA	78 6E	JZ UNZOOM ;YES, STOP ZOOMING	
3351	6EA5	.	.	;FETCH NEW ZOOM PARAMETERS FROM TABLE	
3352	6EA5	3D	.	DCR A ;WANT 0-14,NO ENTRY FOR 1:1	
3353	6EA6	87	.	ADD A ;MULTIPLY BY 8--8 ENTRIES	
3354	6EA7	87	.	ADD A	
3355	6EA8	87	.	ADD A	
3356	6EA9	5F	.	MOV E,A	
3357	6EAA	16	00	MVI D,0 ;DE = INDEX	
3358	6EAC	21	F0 6E	LXI H,ZOOMTB ;BASE OF PARAMETER TABLE	
3359	6EAF	19	.	DAD D ;POINTER TO FIRST PARAM	
3360	6EB0	7E	.	MOV A,M ;FETC 360/MAGNIFICATION	
3361	6EB1	32	EA FB	STA P360M ;STORE +360/MAG LSBYTE	
3362	6EB4	23	.	INX H	
3363	6EB5	7E	.	MOV A,M ;FETCH 180/MAG	
3364	6EB6	32	E8 FB	STA P180M ;STORE +180/MAG LSBYTE	
3365	6EB9	23	.	INX H	
3366	6EBA	7E	.	MOV A,M	
3367	6EBB	32	E6 FB	STA M360M ;STORE -360/MAG LSBYTE	
3368	6EBE	23	.	INX H	
3369	6EBF	7E	.	MOV A,M	
3370	6EC0	32	E4 FB	STA M180M ;STORE -180/MAG LSBYTE	
3371	6EC3	23	.	INX H	
3372	6EC4	7E	.	MOV A,M	
3373	6EC5	32	E2 FB	STA DCBYTE ;STORE DISPLAY CONTROL BYTE	
3374	6EC8	23	.	INX H	
3375	6EC9	7E	.	MOV A,M	
3376	6ECA	32	E3 FB	STA MAXSPD ;STOTE MAX CURSOR SPEED	
3377	6ECD	23	.	INX H ;FETCH LSBYTE OF OTHER	
3378	6ECE	7E	.	MOV A,M ;VERSION OF +180/M	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
3379	6ECF	32	DE FA	STA P180M2	95
3380	6ED2	23	.	INX H ;FETCH OTHER VERSION OF	
3381	6ED3	7E	.	MOV A,M ; -360/M	
3382	6ED4	32	E0 FA	STA M360M2	
3383	6ED7	AF	.	XRA A	
3384	6ED8	32	EB FB	STA P360M+1 ;SEND MSBYTE 360/M	
3385	6ED8	32	E9 FB	STA P180M+1 ;SEND MSBYTE 180/M	
3386	6EDE	32	DF FA	STA P180M2+1	
3387	6EE1	3D	.	DCR A ;A = -1	
3388	6EE2	32	E7 FB	STA M360M+1 ;STORE MSBYTE -360/M	
3389	6EE5	32	E5 FB	STA M180M+1 ;STORE MSBYTE -180/M	
3390	6EE8	32	E1 FA	STA M360M2+1	
3391	6EEB	3E	80 .	MVI A,NWZOOM ;SET NEW ZOOM FLAG	
3392	6EED	C3	40 A2	JMP STFLG5	
3393	6EF0	.	.		
3394	6EF0	.	.	; ZOOMTB EQU \$	
3395	6EF0	B4	5A 4C	DB 180,90,1140,2460,3570 ;2	
3396	6EF5	10	59 4D	DB 200,89,1150	
3397	6EF8	78	3C 88	DB 120,60,2100,3040,3360 ;3	
3398	6EFD	10	3B 89	DB 200,59,2110	
3399	6F00	5A	2D A6	DB 090,45,2460,3230,3150 ;4	
3400	6F05	08	2C A7	DB 100,44,2470	
3401	6F08	48	24 B8	DB 072,36,2700,3340,2740 ;5	
3402	6F0D	08	23 B9	DB 100,35,2710	
3403	6F10	3C	1E C4	DB 060,30,3040,3420,2530 ;6	
3404	6F15	08	1D C5	DB 100,29,3050	
3405	6F18	33	1A CD	DB 051,26,3150,3460,2320 ;7	
3406	6F1D	07	19 CD	DB 070,25,3150	
3407	6F20	2D	17 D3	DB 045,23,3230,3510,2110 ;8	
3408	6F25	07	15 D4	DB 070,21,3240	
3409	6F28	28	14 D8	DB 040,20,3300,3540,1700 ;9	
3410	6F2D	07	13 D9	DB 070,19,3310	
3411	6F30	24	12 DC	DB 036,18,3340,3560,1470 ;10	
3412	6F35	07	11 DD	DB 070,17,3350	
3413	6F38	21	10 DF	DB 033,16,3370,3600,1260 ;11	
3414	6F3D	07	10 E0	DB 070,16,3400	
3415	6F40	1E	0F E2	DB 030,15,3420,3610,1050 ;12	
3416	6F45	07	0E E3	DB 070,14,3430	
3417	6F48	1C	0E E4	DB 028,14,3440,3620,0640 ;13	
3418	6F4D	07	0D E5	DB 070,13,3450	
3419	6F50	1A	0D E6	DB 026,13,3460,3630,0430 ;14	
3420	6F55	07	0C E7	DB 070,12,3470	
3421	6F58	18	0C E8	DB 024,12,3500,3640,0220 ;15	
3422	6F5D	07	0B E9	DB 070,11,3510	
3423	6F60	17	0B E9	DB 023,11,3510,3650,0010 ;16	
3424	6F65	07	0B EB	DB 070,11,3530	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
					96
3426	6F68	.	.	;*****	
3427	6F68	.	.	; ZPOS--SET ZOOM POSITION	
3428	6F68	.	.	; MOVE CURSOR TO DESIGNATED SPOT	
3429	6F68	.	.	; SET NEW ZOOM TO USE CURSOR AS CENTER OF ZOOM	
3430	6F68	.	.	; AREA	
3431	6F68	.	.	;*****	
3432	6F68	.	.	ZPOS EQU \$	
3433	6F68	3E	02	MVI A,2 ;HAVE 2 PARAMETERS?	
3434	6F6A	CD	62	CALL PRMCNT	
3435	6F6D	CC	73	CZ ZPOS1 ;IGNORE IF NOT	
3436	6F70	C3	C1	JMP GEXIT	
3437	6F73	.	.	ZPOS1 EQU \$	
3438	6F73	3E	80	MVI A,NWZOOM ;SET NEW ZOOM	
3439	6F75	CD	40	CALL STFLG5	
3440	6F78	CD	8F	CALL GCP1 ;SET CURSOR POSTION	
3441	6F7B	.	.	; FALL INTO WAIT FOR END OF FRAME	
3442	6F7B	.	.	;*****	
3443	6F7B	.	.	; EOFRM--WAIT FOR END OF FRAME, DO ZOOM, CURSOR	
3444	6F7B	.	.	; UPDATES	
3445	6F7B	.	.	;*****	
3446	6F7B	.	.	EOFRM EQU \$	
3447	6F7B	CD	92	CALL VRWAIT ;WAIT FOR END OF FRAME	
3448	6F7E	C3	CB	JMP VR ;DO VERTICAL RETRACE STUFF	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 97
3450	6F81	. . .	;*****	
3451	6F81	. . .	; AGCPOS--SET CURSOR POSITION FROM ABSOLUTE	
3452	6F81	. . .	; PARAMETER RECEIVED	
3453	6F81	. . .	;*****	
3454	6F81	. . .	AGCPOS EQU \$	
3455	6F81	3E 02 .	MVI A,2 ;WERE 2 VALUES RECEIVED?	
3456	6F83	CD 62 6D	CALL PRMCNT	
3457	6F86	C2 C1 99	JNZ GEXIT ;NO, IGNORE	
3458	6F89	CD 8F 6F	CALL GCP1	
3459	6F8C	C3 C1 99	JMP GEXIT	
3460	6F8F	. . .	GCP1 EQU \$;(INTERNAL ENTRY)	
3461	6F8F	2A B9 90	LHLD PRMBUF ;GET X COORD	
3462	6F92	CD 60 A3	CALL XCHECK ;INSURE IN BOUNDS	
3463	6F95	22 CF 90	SHLD NEWGCX ;STORE NEW X COORD	
3464	6F98	2A BB 90	LHLD PRMBUF+2 ;GET Y COORD	
3465	6F98	CD 69 A3	CALL YCHECK ;INSURE IN BOUNDS	
3466	6F9E	22 CD 90	SHLD NEWGCY ;STORE NEW Y COORD	
3467	6FA1	. . .	GCP2 EQU \$	
3468	6FA1	3E 61 .	MVI A,GCM1+GCM3+GCM4	
3469	6FA3	C3 40 A2	JMP STFLG5 ;SET CURSOR HAS MOVED FLAGS	
3470	6FA6	. . .	;*****	
3471	6FA6	. . .	; IGCP0S--SET CURSOR POSITION FROM INCREMENTAL	
3472	6FA6	. . .	; PARAMETER	
3473	6FA6	. . .	;*****	
3474	6FA6	. . .	IGCP0S EQU \$	
3475	6FA6	3E 02 .	MVI A,2 ;WERE 2 VALUES RECEIVED?	
3476	6FA8	CD 62 6D	CALL PRMCNT	
3477	6FA8	C2 C1 99	JNZ GEXIT ;IGNIRE IF NOT	
3478	6FAE	2A B9 90	LHLD PRMBUF ;GET DELTA X	
3479	6FB1	EB . .	XCHG	
3480	6FB2	2A CF 90	LHLD NEWGCX ;X CURSOR COORD	
3481	6FB5	19 . .	DAD D ;HL = X + INC	
3482	6FB6	CD 60 A3	CALL XCHECK ;INSURE IN BOUNDS	
3483	6FB9	22 CF 90	SHLD NEWGCX ;STORE NEW X COORD	
3484	6FBC	2A BB 90	LHLD PRMBUF+2 ;GET DELTA Y	
3485	6FBF	EB . .	XCHG	
3486	6FC0	2A CD 90	LHLD NEWGCY ;CURSOR Y COORD	
3487	6FC3	19 . .	DAD D ;HL=Y + INC	
3488	6FC4	CD 69 A3	CALL YCHECK ;INSURE IN BOUNDS	
3489	6FC7	22 CD 90	SHLD NEWGCY	
3490	6FCA	CD A1 6F	CALL GCP2 ;SET CURSOR MOVED FLAGS	
3491	6FCD	C3 C1 99	JMP GEXIT	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
					98
3493	6FD0	.	.	*****	
3494	6FD0	.	.	; TGCON--USER WANT GRAPHICS CURSOR TURNED ON	
3495	6FD0	.	.	*****	
3496	6FD0	.	.	TGCON EQU \$	
3497	6FD0	CD	D6 6F	CALL TGCON1	
3498	6FD3	C3	C1 99	JMP GEXIT	
3499	6FD6	.	.	TGCON1 EQU \$;(INTERNAL ENTRY)	
3500	6FD6	3E	01 .	MVI A,SUPRO ;CLEAR SUPRESS FLAG	
3501	6FD8	CD	39 A2	CALL CLFLG3	
3502	6FDB	B7	.	ORA A ;IS CURSOR ALREADY ON?	
3503	6FDC	F8	.	RM ;YES--DONE	
3504	6FDD	F6	80 .	OR1 WANTGC ;NO--SET USER-WANTS-CURSOR-	
3505	6FDF	77	.	MOV M,A ;FLAG, AND STORE	
3506	6FE0	.	.	; IF IN GIN MODE, DONT MOVE CURSOR	
3507	6FE0	3A	AD 90	LDA TKFLGS	
3508	6FE3	E6	10 .	ANI GINMOD ;IN GIN MODE?	
3509	6FE5	C2	F0 6F	JNZ TGC010 ;YES, DONT CHANGE GC LOC	
3510	6FE8	.	.	; IF TURN CURSOR ON WHILE IN TEXT MODE, PUT	
3511	6FE8	.	.	; CURSOR AT CURRENT POINT	
3512	6FE8	3A	97 90	LDA GFLGS6 ;IN GTEXT MODE?	
3513	6FEB	E6	02 .	ANI GTEXT	
3514	6FED	C4	E6 A3	CNZ MOVEGC ;YES, MOVE CURSOR TO PEN	
3515	6FF0	.	.	TGC010 EQU \$	
3516	6FF0	CD	88 9E	CALL GRCON ;TURN THE CURSOR ON	
3517	6FF3	C3	74 A2	JMP SNDGCF ;SEND FLAGS TO DRAW	
3518	6FF6	.	.	*****	
3519	6FF6	.	.	; TGCOFF--USER WANTS GRAPHICS CURSOR TURNED OFF	
3520	6FF6	.	.	*****	
3521	6FF6	.	.	TGCOFF EQU \$	
3522	6FF6	CD	02 70	CALL TGCOF1	
3523	6FF9	C3	C1 99	JMP GEXIT	
3524	6FFC	.	.	*****	
3525	6FFC	.	.	; ROM BREAK 2	
3526	6FFC	.	.	ORG ZBRK1+40000	
3527	7000	.	.	ZBRK2 EQU \$	
3528	7000	54	.	DB VERSN	
3529	7001	70	.	DB ZBRK2/256	
3530	7002	.	.	*****	
3531	7002	.	.	TGCOF1 EQU \$;(INTERNAL ENTRY)	
3532	7002	CD	DE 9E	CALL GRCOFF ;TURN CURSOR OFF	
3533	7005	CD	74 A2	CALL SNDGCF ;SEND GC FLAGS TO DRAW	
3534	7008	CD	87 A2	CALL WAIT ;WAIT FOR CURSOR TO FINISH	
3535	7008	3E	80 .	MVI A,WANTGC ;CLEAR USER-WANTS-CURSOR-FLA	
3536	700D	CD	39 A2	CALL CLFLG3	
3537	7010	C3	4E 70	JMP TRBOF1 ;INSURE RB LINE OFF	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 99
=====
3539      7013      . . .      ;*****
3540      7013      . . .      ; ACON--TURN A/N CURSOR ON
3541      7013      . . .      ;*****
3542      7013      . . .      ACON      EQU      $
3543      7013      CD 19 70      CALL ACON1
3544      7016      C3 C1 99      JMP      GEXIT
3545      7019      . . .      ACON1     EQU      $      ;(INTERNAL ENTRY)
3546      7019      3E 40 .      MVI      A,ACINHB ;CLEAR INHIBIT BIT
3547      7018      CD 2C A2      CALL CLFLG1
3548      701E      3A C0 FF      LDA      ZCUROW   ;FETCH CURRENT CURSOR ROW
3549      7021      C3 B7 A2      JMP      ANCHK    ;PUT IT THERE IF A/N ON
3550      7024      . . .      ;*****
3551      7024      . . .      ; ACOFF--TURN A/N CURSOR OFF
3552      7024      . . .      ;*****
3553      7024      . . .      ACOFF     EQU      $
3554      7024      3E 40 .      MVI      A,ACINHB ;SET INHIBIT BIT
3555      7026      CD 26 A2      CALL STFLG1
3556      7029      3E 18 .      MVI      A,ZMXROW+1 ;PUT CURSOR OFF VISIBLE
3557      702B      32 20 87      STA      ZIOCRW   ;DISPLAY
3558      702E      C3 C1 99      JMP      GEXIT
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 100
3560	7031	.	.	. ;*****	
3561	7031	.	.	. ; TRBON--TURN RUBBER BAND LINE ON	
3562	7031	.	.	. ; ALSO TURN CURSOR ON IF NOT ALREADY ON	
3563	7031	.	.	. ; RB LINE ACTUALLY GOES ON IN VR ROUTINE	
3564	7031	.	.	. ;*****	
3565	7031	.	.	. TRBON EQU \$	
3566	7031	CD	37 70	CALL TRBON1	
3567	7034	C3	C1 99	JMP GEXIT	
3568	7037	.	.	. TRBON1 EQU \$;(INTERNAL ENTRY)	
3569	7037	CD	F4 88	CALL CHEKAP ;DONT TURN ON IF AUTO PLOT ON	
3570	703A	C0	.	. RNZ	
3571	703B	3E	20 .	MVI A,WANTRB	
3572	703D	CD	33 A2	CALL STFLG3 ;SET WANT RBLINE FLAG	
3573	7040	3E	20 .	MVI A,GCM3 ;FAKE A CURSOR MOVE TO INSUR	
3574	7042	CD	40 A2	CALL STFLG5 ;IT GOES ON	
3575	7045	C3	D6 6F	JMP TGCON1 ;TURN THE CURSOR ON	
3576	7048	.	.	. ;*****	
3577	7048	.	.	. ; TRBOFF--TURN RUBBER BAND LINE OFF	
3578	7048	.	.	. ;*****	
3579	7048	.	.	. TRBOFF EQU \$	
3580	7048	CD	4E 70	CALL TRBOF1	
3581	704B	C3	C1 99	JMP GEXIT	
3582	704E	.	.	. TRBOF1 EQU \$;(INTERNAL ENTRY)	
3583	704E	3E	20 .	MVI A,WANTRB ;CLEAR WANT RB LINE FLAG	
3584	7050	CD	39 A2	CALL CLFLG3	
3585	7053	C3	7E 70	JMP RBOFF ;TURN IT OFF	
3586	7056	.	.	. ;*****	
3587	7056	.	.	. ; RBON--DRAW RUBBER BAND LINE TO NEW CURSOR LOC	
3588	7056	.	.	. ;*****	
3589	7056	.	.	. RBON EQU \$	
3590	7056	3A	B0 90	LDA GFLG3 ;IS IT ALREADY ON? SUPRESSED	
3591	7059	E6	1F .	ANI RBISON+SUPR0+SUPR1+SUPR2+TIMSUP	
3592	705B	C0	.	. RNZ ;YES, LEAVE AS IS	
3593	705C	CD	20 A2	CALL NORST ;DISALLOW RESETS	
3594	705F	3E	10 .	MVI A,RBISON ;SET ACTUALLY ON FLAG	
3595	7061	CD	33 A2	CALL STFLG3	
3596	7064	3E	10 .	MVI A,RBDRW	
3597	7066	CD	40 A2	CALL STFLG5 ;SET 'DRAWING RBLINE' FLAG	
3598	7069	2A	CD 90	LHLD NEWGXY ;LOAD CURSOR COORDINATES	
3599	706C	22	7D 90	SHLD RBY ;SAVE RBLINE COORDINATES	
3600	706F	EB	.	. XCHG	
3601	7070	2A	CF 90	LHLD NEWGCX	
3602	7073	22	7F 90	SHLD RBX	
3603	7076	.	.	. RB0010 EQU \$	
3604	7076	CD	9B 70	CALL DRAWRB ;COMPLEMENT THE RB LINE	
3605	7079	3E	10 .	MVI A,RBDRW ;CLEAR 'DRAWING RBLINE'	
3606	707B	C3	46 A2	JMP CLFLG5 ;FLAG	


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 101
=====
3608      707E      .      .      .      ;*****
3609      707E      .      .      .      ; RBOFF--UNDRAW RUBBER BAND LINE TO CURRENT
3610      707E      .      .      .      ; CURSOR LOCATION
3611      707E      .      .      .      ;*****
3612      707E      .      .      .      RBOFF EQU $
3613      707E      3A      B0      90      LDA GFLGS3 ;IS IT ALREADY OFF?
3614      7081      E6      10      .      ANI RBISON
3615      7083      C8      .      .      RZ ;YES--DONE
3616      7084      CD      20      A2      CALL NORST ;(DISALLOW RESETS)
3617      7087      3E      10      .      MVI A,RBISON
3618      7089      CD      39      A2      CALL CLFLG3 ;NO--CLEAR 'ON' FLAG
3619      708C      3E      10      .      MVI A,RBDRW ;SET 'DRAWING RBLINE' FLAG
3620      708E      CD      40      A2      CALL STFLG5
3621      7091      2A      7D      90      LHLD RBY ;LOAD RB LINE COORDS
3622      7094      EB      .      .      XCHG
3623      7095      2A      7F      90      LHLD RBX
3624      7098      C3      76      70      JMP RBO010 ;ERASE THE RB LINE
=====
  
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 102
3626	709B	. . .	;*****	
3627	709B	. . .	; DRAWRB--DRAW RUBBER BAND LINE FROM CURRENT	
3628	709B	. . .	; POINT TO GRAPHICS CURSOR IN COMPLEMENT MODE	
3629	709B	. . .	; ENTRY HL = CURSOR X COORD	
3630	709B	. . .	; DE = CURSOR Y COORD	
3631	709B	. . .	;*****	
3632	709B	. . .	DRAWRB EQU \$	
3633	709B	4D . .	MOV C,L ;SAVE HL	
3634	709C	44 . .	MOV B,H	
3635	709D	. . .	; SAVE ALL VARIABLES THAT WOULD BE USED TO	
3636	709D	. . .	; DRAW VECTOR IN CASE THIS IS CALLED FROM	
3637	709D	. . .	; VECTOR ROUTINE	
3638	709D	2A DA 90	LHLD XNEW ;SAVE NEW POINT	
3639	70A0	E5 . .	PUSH H	
3640	70A1	2A DB 90	LHLD YNEW	
3641	70A4	E5 . .	PUSH H	
3642	70A5	69 . .	MOV L,C ;RESTORE HL	
3643	70A6	60 . .	MOV H,B	
3644	70A7	22 DA 90	SHLD XNEW ;STORE X COORD	
3645	70AA	EB . .	XCHG	
3646	70AB	22 DB 90	SHLD YNEW ;STORE Y COORD	
3647	70AE	. . .	; SAVE CURRENT POINT AND RELATED VARIABLES	
3648	70AE	2A DE 90	LHLD XCURR	
3649	70B1	E5 . .	PUSH H	
3650	70B2	2A DC 90	LHLD YCURR	
3651	70B5	E5 . .	PUSH H	
3652	70B6	3A D2 90	LDA CURCD ;BOUNDS CODE	
3653	70B9	F5 . .	PUSH PSW	
3654	70BA	3A B2 90	LDA GFLGS1 ;DRAW/MOVE FLAG	
3655	70BD	F5 . .	PUSH PSW	
3656	70BE	E6 FE .	ANI -1-MOVE ;CLEAR THE MOVE FLAG	
3657	70C0	32 B2 90	STA GFLGS1	
3658	70C3	3A B5 90	LDA CURMOD ;CURRENT DRAWING MODE	
3659	70C6	F5 . .	PUSH PSW	
3660	70C7	E6 10 .	ANI GVENAB ;ONLY NEED VIDEO BIT	
3661	70C9	F6 03 .	ORI 30 ;PUT INTO COMPLEMENT MODE	
3662	70CB	32 B5 90	STA CURMOD	
3663	70CE	3A AE 90	LDA GFLGS5	
3664	70D1	F5 . .	PUSH PSW ;SAVE GFLGS5	
3665	70D2	F6 08 .	ORI DWFRST ;SET DRAW FIRST DOT	
3666	70D4	32 AE 90	STA GFLGS5 ;FLAG	
3667	70D7	3A 97 90	LDA GFLGS6 ;SAVE TEXT FLAGS	
3668	70DA	F5 . .	PUSH PSW	
3669	70DB	F6 04 .	ORI NOSOL ;DONT UPDATE SOL	
3670	70DD	32 97 90	STA GFLGS6	
3671	70E0	CD EC 65	CALL VECTR1 ;DRAW FROM CURPT TO CURSOR	
3672	70E3	F1 . .	POP PSW ;RESTORE TEXT FLAGS	
3673	70E4	32 97 90	STA GFLGS6	
3674	70E7	F1 . .	POP PSW ;RECALL GFLGS5	
3675	70E8	32 AE 90	STA GFLGS5	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS
3676	70EB	F1	.	.	POP PSW
3677	70EC	32	B5	90	STA CURMOD ;RESTORE DRAWING MODE
3678	70EF	F1	.	.	POP PSW
3679	70F0	F6	08	.	ORI NEWWA ;MUST RECOMPUTE WA
3680	70F2	32	B2	90	STA GFLGS1
3681	70F5	F1	.	.	POP PSW
3682	70F6	32	D2	90	STA CURCD ;RESTORE BOUNDS CODE
3683	70F9	E1	.	.	POP H
3684	70FA	22	DC	90	SHLD YCURR ;RESTORE CURRENT POINT
3685	70FD	E1	.	.	POP H
3686	70FE	22	DE	90	SHLD XCURR
3687	7101	E1	.	.	POP H ;RESTORE NEW POINT
3688	7102	22	D8	90	SHLD YNEW
3689	7105	E1	.	.	POP H
3690	7106	22	DA	90	SHLD XNEW
3691	7109	C9	.	.	RET

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 104
=====
3693     710A      . . .      ;*****
3694     710A      . . .      ; MODSEQ--ESC * M RECEIVED, SET UP FOR MODE
3695     710A      . . .      ; CONTROL ESCAPE SEQUENCE
3696     710A      . . .      ;*****
3697     710A      . . .      MODSEQ EQU $
3698     710A      21 12 61    LXI H,MODTAB ;SET MODE CONTROL TABLE AS
3699     710D      C3 86 63    JMP SETRTB
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
3701	7110	.	.	;*****
3702	7110	.	.	; ABFILL FILL RECTANGULAR AREA SPECIFIED BY
3703	7110	.	.	; LOWER LEFT, UPPER RIGHT POINT WITH HORIZONTAL
3704	7110	.	.	; VECTORS USING CURRENT DRAWING MODE
3705	7110	.	.	; VECTORS ARE DRAWN LEFT TO RIGHT, BOTTOM TO TOP
3706	7110	.	.	; DOES NOT CHANGE CURRENT POINT
3707	7110	.	.	;*****
3708	7110	.	.	ABFILL EQU \$
3709	7110	3E	04	MVI A,4 ;GOT FOUR PARAMETERS?
3710	7112	CD	62 6D	CALL PRMCNT
3711	7115	C2	C1 99	JNZ GEXIT ;IGNORE IF NOT
3712	7118	.	.	; SUPPRESS CURSOR NOW, BOTH CURSOR AND AREA FILL
3713	7118	.	.	; USE TEMP VARIABLES
3714	7118	3E	02	MVI A,SUPR1
3715	711A	CD	8A 9E	CALL SUPRGC ;SUPPRESS CURSOR
3716	711D	.	.	;CONVERT LL,UR POINTS TO INTERNAL FORMAT
3717	711D	2A	B9 90	LHLD PRMBUF ;GET X COORD, LL
3718	7120	CD	60 A3	CALL XCHECK ;CHECK IF IN BOUNDS
3719	7123	22	69 90	SHLD XLEFT ;SAVE X COORD, LOWER LEFT
3720	7126	2A	BB 90	LHLD PRMBUF+2 ;GET Y COORD, LL
3721	7129	CD	69 A3	CALL YCHECK ;INSURE IN BOUNDS
3722	712C	22	67 90	SHLD YBOT ;SAVE Y COORD, LOWER LEFT
3723	712F	2A	BD 90	LHLD PRMBUF+4 ;GET X COORD, UR
3724	7132	CD	60 A3	CALL XCHECK ;INSURE X IN BOUNDS
3725	7135	E5	.	PUSH H ;SAVE X COORD, UPPER RIGHT
3726	7136	2A	BF 90	LHLD PRMBUF+6 ;GET Y COORD, UR
3727	7139	CD	69 A3	CALL YCHECK ;INSURE Y IN BOUNDS
3728	713C	EB	.	XCHG ;DE = Y COORD, UPPER RIGHT
3729	713D	.	.	;ALTERNATE ENTRY FOR RELATIVE AREA FILL
3730	713D	.	.	AFILL1 EQU \$
3731	713D	.	.	; COMPUTE HEIGHT OF RECTANGLE
3732	713D	2A	67 90	LHLD YBOT ;COMPUTE HEIGHT = YTOP-YBOT
3733	7140	CD	09 A3	CALL NEGATE ;HL = -YBOT
3734	7143	19	.	DAD D ;HL = YTOP-YBOT
3735	7144	.	.	; IF VALUE IS -, BOTTOM WAS GREATER THAN TOP
3736	7144	.	.	; EXIT NOW
3737	7144	7C	.	MOV A,H ;CHECK SIGN OF HEIGHT
3738	7145	B7	.	ORA A ;IS IT -?
3739	7146	FA	CB 71	JM AF015 ;PARAMETERS WERE BAD, EXIT
3740	7149	22	63 90	SHLD HEIGHT ;HL = # OF VECTORS TO DRAW
3741	714C	.	.	; COMPUTE LENGTH OF RECTANGLE = LENGTH OF VECTORS
3742	714C	2A	69 90	LHLD XLEFT ;COMPUTE -LEN = -(XRT-XLEFT)
3743	714F	EB	.	XCHG ;DE = XLEFT
3744	7150	E1	.	POP H ;HL = XRIGHT
3745	7151	CD	09 A3	CALL NEGATE ;HL = - XRIGHT
3746	7154	19	.	DAD D ;HL = XLEFT-XRIGHT
3747	7155	2B	.	DCX H ;HL = -VECTOR LENGTH FOR HW
3748	7156	.	.	; IF VALUE IS +, XRIGHT WAS > XLEFT, EXIT NOW
3749	7156	7C	.	MOV A,H ;CHECK SIGN OF LENGTH
3750	7157	B7	.	ORA A ;IS IT +?

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 106
3751	7158	F2	CC	71	JP AF020 ;YES,PARAMETERS BAD, EXIT	
3752	7158	E5	.	.	PUSH H ;SAVE LENGTH TO SEND TO HW	
3753	715C	.	.	.	; CONVERT STARTING ADDRESS TO SCREEN COORDS	
3754	715C	2A	67	90	LHLD YBOT	
3755	715F	CD	5B	67	CALL MPY45 ;COMPUTE 45 * Y	
3756	7162	22	65	90	SHLD YBOT45	
3757	7165	.	.	.	; FETCH FIRST PATTERN BYTE IF AREA PATTERN ON	
3758	7165	3A	B2	90	LDA GFLGS1 ;IS AREA PATTERN ON?	
3759	7168	E6	04	.	ANI AREAPT	
3760	716A	C4	FD	A3	CNZ GTPAT2 ;IF YES, GET FIRST PAT BYTE	
3761	716D	.	.	.	; LOAD CONTROLLER WITH CONSTANT PARAMETERS FOR HOR	
3762	716D	.	.	.	; IZONTAL VECTOR. HARDWARE IS IDLE FROM CURSOR	
3763	716D	.	.	.	; SUPRESS	
3764	716D	E1	.	.	POP H ;RECALL VECTOR LENGTH	
3765	716E	22	12	89	SHLD DC ;SEND DOT COUNT	
3766	7171	CD	0F	A4	CALL VSETUP ;SET CONSTANT PARAMETERS	
3767	7174	32	18	89	STA SIGNM1 ;SET M1 = 1 FOR 1ST OCTANT	
3768	7177	23	.	.	INX H	
3769	7178	22	1A	89	SHLD M1 ;SET M1 = +1	
3770	717B	.	.	.	; SEND SCALE FACTOR	
3771	717B	3A	B3	90	LDA SCALE	
3772	717E	32	21	89	STA SCALER ;SEND TO HW	
3773	7181	.	.	.	;SEND DRAWING MODE	
3774	7181	3A	B5	90	LDA CURMOD	
3775	7184	32	41	89	STA HCEJK	
3776	7187	.	.	.	;MAIN LOOP TO SEND VECTORS	
3777	7187	.	.	.	AF000 EQU \$	
3778	7187	.	.	.	;SEND NEW WRITE ADDRESS	
3779	7187	2A	69	90	LHLD XLEFT ;HL = X COORD FOR ALL VECTOR	
3780	718A	EB	.	.	XCHG	
3781	718B	2A	65	90	LHLD YBOT45 ;HL = Y * 45	
3782	718E	CD	6F	67	CALL GETWA ;CONVERT TO WA	
3783	7191	22	0E	89	SHLD LSBWA ;SEND 12 LSBITS	
3784	7194	32	0C	89	STA MSBWA ;SEND 6 MSBITS	
3785	7197	.	.	.	;SEND PATTERN	
3786	7197	3A	B4	90	LDA CURPAT	
3787	719A	32	40	89	STA PATERN	
3788	719D	.	.	.	;DRAW THE VECTOR	
3789	719D	CD	22	67	CALL HWGO ;START HW GOING	
3790	71A0	.	.	.	;SEE IF ALL OF THE VECTORS HAVE BEEN SENT	
3791	71A0	2A	63	90	LHLD HEIGHT ;RECALL THE HEIGHT	
3792	71A3	2B	.	.	DCX H	
3793	71A4	7C	.	.	MOV A,H ;SEE IF DOWN TO 0 YET	
3794	71A5	B7	.	.	ORA A	
3795	71A6	FA	CC	71	JM AF020 ;YES--DONE	
3796	71A9	22	63	90	SHLD HEIGHT ;NO--UPDATE HEIGHT	
3797	71AC	.	.	.	;UPDATE WRITE ADDRESS--MOVE UP ONE LINE	
3798	71AC	2A	65	90	LHLD YBOT45 ;WA IN SCREEN COORDS	
3799	71AF	11	D3	FF	LXI D,-45 ;GO UP ONE LINE	
3800	71B2	19	.	.	DAD D ;ADD 45 TO CURRENT LINE	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 107
3801	71B3	22	65	90	SHLD YBOT45
3802	71B6	.	.	.	;IF AREA PATTERN ON, FETCH NEXT PATTERN
3803	71B6	3A	B2	90	LDA GFLGS1
3804	71B9	E6	04	.	ANI AREAPT ;AREA PATTERN ON?
3805	71B8	CA	C5	71	JZ AF010 ;NO
3806	71BE	21	67	90	LXI H,YBOT ;YES--UPDATE Y COORD TO
3807	71C1	34	.	.	INR M ;SELECT PROPER PATTERN
3808	71C2	.	.	.	;USING XLEFT AND YBOT, SELECT PROPER PATTERN
3809	71C2	.	.	.	;AND ROTATE INTO POSITION. ONLY NEED 3 LSBITS.
3810	71C2	CD	FD	A3	CALL GTPAT2
3811	71C5	.	.	.	AF010 EQU \$
3812	71C5	CD	87	A2	CALL WAIT ;WAIT FOR HW TO FINISH
3813	71C8	C3	87	71	JMP AF000 ;DO THE NEXT ONE
3814	71CB	.	.	.	AF015 EQU \$
3815	71CB	E1	.	.	POP H ;CLEAN UP STACK (ERROR)
3816	71CC	.	.	.	AF020 EQU \$
3817	71CC	.	.	.	;AREA FILL DONE, CLEAN THINGS UP
3818	71CC	3E	08	.	MVI A,NEWWA ;SET USE NEW WA FLAG
3819	71CE	CD	26	A2	CALL STFLG1
3820	71D1	3E	02	.	MVI A,SUPR1 ;UNSUPRESS THE CURSOR
3821	71D3	CD	B5	9E	CALL ENABGC
3822	71D6	C3	C1	99	JMP GEXIT ;DONE
3823	71D9	.	.	.	;*****
3824	71D9	.	.	.	; RLFILL--AREA FILL USING RELOCATABLE COORDINATES
3825	71D9	.	.	.	; SAME AS ABSOLUTE FILL, BUT ADD VALUE OF
3826	71D9	.	.	.	; RELOCATABLE ORIGIN TO COORDS
3827	71D9	.	.	.	;*****
3828	71D9	.	.	.	RLFILL EQU \$
3829	71D9	3E	04	.	MVI A,4 ;HAVE 4 VALUES?
3830	71DB	CD	62	6D	CALL PRMCNT
3831	71DE	C2	C1	99	JNZ GEXIT ;NO, IGNORE COMMAND
3832	71E1	3E	01	.	MVI A,1
3833	71E3	CD	8A	9E	CALL SUPRGC ;SUPRESS THE CURSOR
3834	71E6	2A	9A	90	LHLD XORG
3835	71E9	EB	.	.	XCHG ;DE = X RELOC ORIGIN VALUE
3836	71EA	2A	B9	90	LHLD PRMBUF ;FETCH LOWER LEFT X COORD
3837	71ED	19	.	.	DAD D ;ADD RELOC ORG
3838	71EE	CD	60	A3	CALL XCHECK ;INSURE IN BOUNDS
3839	71F1	22	69	90	SHLD XLEFT ;STORE X COORD
3840	71F4	2A	BD	90	LHLD PRMBUF+4 ;FETCH UPPER RIGHT X COORD
3841	71F7	19	.	.	DAD D ;ADD RELOC ORIG
3842	71F8	CD	60	A3	CALL XCHECK ;INSURE IN BOUNDS
3843	71FB	E5	.	.	PUSH H ;SAVE ON STACK
3844	71FC	2A	98	90	LHLD YORG
3845	71FF	EB	.	.	XCHG ;DE = Y RELOC ORG VALUE
3846	7200	2A	BB	90	LHLD PRMBUF+2 ;FETCH Y COORD, LOWER LEFT
3847	7203	19	.	.	DAD D ;ADD RELOC ORG
3848	7204	CD	69	A3	CALL YCHECK ;INSURE INBOUNDS
3849	7207	22	67	90	SHLD YBOT ;STORE Y COORD
3850	720A	2A	BF	90	LHLD PRMBUF+6 ;FETCH Y COORD, UPPER RIGHT

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 108
=====
3851      720D      19      .      .      DAD      D      ;ADD RELOC ORG
3852      720E      CD      69      A3     CALL      YCHECK ;INSURE IN BOUNDS
3853      7211      EB      .      .      XCHG                                     ;LEAVE Y COORD IN DE
3854      7212      C3      3D      71     JMP      AFILL1  ;CONTINUE AS WITH ABSOLUTE
=====
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 109
3856	7215	.	.	*****	
3857	7215	.	.	; SETMOD--SET DRAWING MODE = JK BITS	
3858	7215	.	.	; 00 = DO NOTHING/JAM PATTERN	
3859	7215	.	.	; 01 = CLEAR	
3860	7215	.	.	; 10 = SET	
3861	7215	.	.	; 11 = COMPLEMENT	
3862	7215	.	.	; 100 = JAM PATTERN	
3863	7215	.	.	*****	
3864	7215	.	.	SETMOD EQU \$	
3865	7215	3E	04	MVI A,4 ;MAX VALUE	
3866	7217	CD	43 6D	CALL GETPRM ;FETCH PARAMETER	
3867	721A	CC	20 72	CZ SETMD1 ;IGNORE IF BAD	
3868	721D	C3	C1 99	JMP GEXIT	
3869	7220	.	.	SETMD1 EQU \$;(INTERNAL ENTRY)	
3870	7220	B7	.	ORA A ;DO NOTHING SELECTED?	
3871	7221	CA	28 72	JZ STM010 ;YES, LEAVE EJK = 0	
3872	7224	E6	03	ANI 30 ;WANT JK BITS ONLY	
3873	7226	F6	04	ORI PATENB ;LEAVE PATTERN ON	
3874	7228	.	.	STM010 EQU \$	
3875	7228	4F	.	MOV C,A ;LEAVE EJK BITS IN C	
3876	7229	21	B5 90	LXI H,CURMOD ;FETCH CURRENT MODE	
3877	722C	7E	.	MOV A,M	
3878	722D	E6	F8	ANI 3700 ;DELETE OLD EJK	
3879	722F	B1	.	ORA C ;MERGE IN NEW	
3880	7230	77	.	MOV M,A ;STORE NEW MODE	
3881	7231	C9	.	RET	
3882	7232	.	.	*****	
3883	7232	.	.	; SNDMOD--SEND MODE BITS TO HW	
3884	7232	.	.	; ENTRY A = MODE	
3885	7232	.	.	*****	
3886	7232	.	.	SNDMOD EQU \$	
3887	7232	CD	87 A2	CALL WAIT ;WAIT FOR HW TO BE FREE	
3888	7235	32	41 89	STA HCEJK ;SEND MODE	
3889	7238	C9	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 110
3891	7239	.	.	*****	
3892	7239	.	.	; SETLIN--SET LINE TYPE	
3893	7239	.	.	; 1 = SOLID	
3894	7239	.	.	; 2 = USER DEFINED DOT-DASH	
3895	7239	.	.	; 3 = USER DEFINED AREA	
3896	7239	.	.	; 4-11 = PREDEFINED LINE TYPES	
3897	7239	.	.	; TYPE 11 => POINT PLOT (SINGLE DOT AT ENDPOINT)	
3898	7239	.	.	*****	
3899	7239	.	.	SETLIN EQU \$	
3900	7239	3E	0B	MVI A,11 ;MAX VALUE	
3901	723B	CD	43 6D	CALL GETPRM ;GET THE PARAMETER	
3902	723E	C2	C1 99	JNZ GEXIT ;IGNORE IF BAD	
3903	7241	3D	.	DCR A ;WANT 0-10, NOT 1-11	
3904	7242	F4	48 72	CP SETLN1 ;IGNORE IF 0	
3905	7245	C3	C1 99	JMP GEXIT	
3906	7248	.	.	SETLN1 EQU \$;(INTERNAL ENTRY)	
3907	7248	4F	.	MOV C,A ;STORE LINE TYPE	
3908	7249	3E	08	MVI A,DWFRST ;SET 'DRAW FIRST DOT FLAG'	
3909	724B	CD	40 A2	CALL STFLG5	
3910	724E	3E	06	MVI A,AREAPT+LINEPT ;CLEAR USER DEFINED	
3911	7250	CD	2C A2	CALL CLFLG1 ;AREA/LINE PATTERN FLAGS	
3912	7253	79	.	MOV A,C ;RECALL LINE TYPE	
3913	7254	FE	01	CPI UDLINE ;USER DEFINED LINE PAT?	
3914	7256	CA	7F 72	JZ STL030 ;YES	
3915	7259	FE	02	CPI UDAREA ;USER DEFINED AREA PAT?	
3916	725B	CA	91 72	JZ STL040 ;YES	
3917	725E	.	.	; PROCESS PREDEFINED LINE TYPE	
3918	725E	B7	.	ORA A ;TYPE 0 SELECTED?	
3919	725F	CA	6A 72	JZ STL010 ;YES	
3920	7262	D6	02	SUI 2 ;ADJUST FOR USER DEFINED	
3921	7264	.	.	SETLN2 EQU \$;(ENTRY FROM AUTOPLLOT)	
3922	7264	FE	09	CPI MAXTYP ;TOO BIG? (FROM AUTOPLLOT)	
3923	7266	DA	6A 72	JC STL010 ;NO	
3924	7269	AF	.	XRA A ;YES, USE SOLID	
3925	726A	.	.	STL010 EQU \$	
3926	726A	32	DB FA	STA LNTYPE ;SAVE THE LINE TYPE	
3927	726D	87	.	ADD A ;A * 2	
3928	726E	5F	.	MOV E,A ;USE AS INDEX	
3929	726F	16	00	MVI D,0 ;TO LINE PATTERN TABLE	
3930	7271	21	9B 72	LXI H,LINETB ;BASE OF TABLE	
3931	7274	19	.	DAD D ;INDEX TO PATTERN	
3932	7275	7E	.	MOV A,M ;FETCH PATTERN	
3933	7276	32	B4 90	STA CURPAT	
3934	7279	23	.	INX H ;POINTER TO SCALE FACTOR	
3935	727A	7E	.	MOV A,M	
3936	727B	32	B3 90	STA SCALE	
3937	727E	C9	.	RET	
3938	727F	.	.	STL030 EQU \$;SET USER DEFINED LINE PAT	
3939	727F	3E	02	MVI A,LINEPT ;TURN USER-DEFINED LINE PAT	
3940	7281	CD	26 A2	CALL STFLG1 ;FLAG ON	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 111
=====
3941      7284      3A  A8  90          LDA  LPAT          ;FETCH USER DEFINED PATTERN
3942      7287      32  B4  90          STA  CURPAT
3943      728A      3A  A7  90          LDA  LSCALE
3944      728D      32  B3  90          STA  SCALE
3945      7290      C9  .   .           RET
3946      7291      .   .   .           STL040 EQU $      ;SET USER DEFINED AREA PAT
3947      7291      3E  04  .           MVI  A,AREAPT
3948      7293      CD  26  A2          CALL STFLG1
3949      7296      AF  .   .           XRA  A             ;TURN PRESCALER OFF
3950      7297      32  B3  90          STA  SCALE
3951      729A      C9  .   .           RET
3952      729B      .   .   .           ;
3953      729B      .   .   .           LINET8 EQU $
3954      729B      FF  00  .           DB   377Q,0       ;DEFAULT, SOLID
3955      729D      E4  02  .           DB   344Q,2       ;CENTER LINE
3956      729F      FC  02  .           DB   374Q,2       ;LONG DASH
3957      72A1      E0  01  .           DB   340Q,1       ;SHORT DASH
3958      72A3      AA  00  .           DB   252Q,0       ;HALF BRIGHT
3959      72A5      FA  01  .           DB   372Q,1       ;DASH-DOT-DASH
3960      72A7      54  02  .           DB   124Q,2       ;DOT-DOT-DOT
3961      72A9      D5  03  .           DB   325Q,3       ;DASH-DOT-DOT
3962      72AB      FF  00  .           DB   377Q,0       ;POINT PLOT
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 112
3964	72AD	. . .	;*****	
3965	72AD	. . .	; DEFLP--DEFINE LINE PATTERN	
3966	72AD	. . .	; FIRST PARAMETER = PATTERN	
3967	72AD	. . .	; SECOND = SCALE (1-16)	
3968	72AD	. . .	; NO BOUNDS CHECK ON PATTERN	
3969	72AD	. . .	;*****	
3970	72AD	. . .	DEFLP EQU \$	
3971	72AD	3E 02 .	MVI A,2 ;GOT 2 PARAMETERS	
3972	72AF	CD 62 6D	CALL PRMCNT	
3973	72B2	C2 C1 99	JNZ GEXIT ;NO, IGNORE	
3974	72B5	3A BC 90	LDA PRMBUF+3 ;MSBYTE OF SCALE MUST BE 0	
3975	72B8	B7 . .	ORA A	
3976	72B9	C2 C1 99	JNZ GEXIT ;BAD, IGNORE	
3977	72BC	3A BB 90	LDA PRMBUF+2 ;FETCH SCALE FACTOR	
3978	72BF	FE 11 .	CPI 17 ;.GT. 16?	
3979	72C1	D2 C1 99	JNC GEXIT ;YES, IGNORE	
3980	72C4	3D . .	DCK A ;WANT 0-15, NOT 1-16	
3981	72C5	FA C1 99	JM GEXIT ;IGNORE IF WAS 0	
3982	72C8	32 A7 90	STA LSCALE ;STORE SCALE FACTOR	
3983	72CB	47 . .	MOV B,A ;SAVE IT	
3984	72CC	3A B9 90	LDA PRMBUF ;FETCH PATTERN BYTE	
3985	72CF	32 A8 90	STA LPAT ;STORE LINE PATTERN	
3986	72D2	4F . .	MOV C,A ;SAVE PATTERN	
3987	72D3	. . .	; IF USER-DEFINED LINE PATTERN IS ON NOW, USE	
3988	72D3	. . .	; NEW PATTERN	
3989	72D3	3A B2 90	LDA GFLGS1 ;LINE PATTERN ON?	
3990	72D6	E6 02 .	ANI LINEPT	
3991	72D8	CA C1 99	JZ GEXIT ;NO, DONE	
3992	72DB	21 B3 90	LXI H,SCALE ;YES, STORE NEW SCALE	
3993	72DE	70 . .	MOV M,B	
3994	72DF	21 B4 90	LXI H,CURPAT ;STORE NEW PATTERN	
3995	72E2	71 . .	MOV M,C	
3996	72E3	3E 08 .	MVI A,DWFRST ;SET DRAW FIRST DOT	
3997	72E5	CD 40 A2	CALL STFLGS	
3998	72E8	C3 C1 99	JMP GEXIT	
3999	72EB	. . .	;*****	
4000	72EB	. . .	; DEFAP--DEFINE AREA PATTERN	
4001	72EB	. . .	; 16 LOCATIONS IN PRMBUF = 8 PATTERN BYTES	
4002	72EB	. . .	;*****	
4003	72EB	. . .	DEFAP EQU \$	
4004	72EB	3E 08 .	MVI A,8 ;8 PARAMS RECEIVED?	
4005	72ED	CD 62 6D	CALL PRMCNT	
4006	72F0	C2 C1 99	JNZ GEXIT	
4007	72F3	. . .	; DONT DO BOUNDS CHECK ON PATTERN	
4008	72F3	0E 07 .	MVI C,7 ;LOOP COUNTER FOR 8 BYTES	
4009	72F5	11 F7 FB	LXI D,HAPAT ;BUFFER FOR HORIZ PAT BYTES	
4010	72F8	21 B9 90	LXI H,PRMBUF ;START OF PARAMETERS	
4011	72FB	. . .	DAPU10 EQU \$	
4012	72FB	7E . .	MOV A,M ;FETCH PATTERN	
4013	72FC	12 . .	STAX D ;STORE IT	

=====					PAGE 113	
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	
=====						
4014	72FD	23	.	.	INX H	;SKIP MSBYTE OF PATTERN
4015	72FE	23	.	.	INX H	
4016	72FF	13	.	.	INX D	;UPDATE DEST. POINTER
4017	7300	0D	.	.	DCR C	;DEC LOOP COUNTER
4018	7301	F2	FB	72	JP DAP010	;GO THRU AGAIN
4019	7304	.	.	.	;NOW, REFORMAT HORIZONTAL PATTERN BYTES INTO	
4020	7304	.	.	.	;VERTICAL PATTERN BYTES	
4021	7304	1E	07	.	MVI E,7	;COUNTER FOR 8 NEW BYTES
4022	7306	01	EF	FB	LXI B,VAPAT	;DESTINATION OF NEW BYTES
4023	7309	.	.	.	DAP020 EQU \$	
4024	7309	16	07	.	MVI D,7	;EXAMINE 8 BYTES,1 BIT AT A
4025	7308	21	F7	FB	LXI H,HAPAT	;TIME. SOURCE OF PAT BYTES
4026	730E	.	.	.	DAP030 EQU \$	
4027	730E	7E	.	.	MOV A,M	;FETCH HORIZONTAL BYTE
4028	730F	07	.	.	RLC	;SEE IF BIT ON
4029	7310	77	.	.	MOV M,A	;STORE ROTATED BYTE
4030	7311	0A	.	.	LDAX B	;FETCH BUDDING VERTICIAL BYTE
4031	7312	D2	1A	73	JNC DAP040	;JMP IF BIT OFF
4032	7315	F6	80	.	ORI 2000	;ADD BIT TO VERTICAL PAT
4033	7317	C3	1C	73	JMP DAP050	
4034	731A	.	.	.	DAP040 EQU \$	
4035	731A	E6	7F	.	ANI 1770	;CLEAR BIT OF VERTICAL PAT
4036	731C	.	.	.	DAP050 EQU \$	
4037	731C	07	.	.	RLC	;ROTATE VERTICAL BYTE
4038	731D	02	.	.	STAX B	;STORE VERTICAL BYTE
4039	731E	23	.	.	INX H	;GET NEXT HORIZONTAL BYTE
4040	731F	15	.	.	DCR D	;ANY HORIZ LEFT??
4041	7320	F2	0E	73	JP DAP030	;YES--GO THROUGH AGAIN
4042	7323	03	.	.	INX B	;NEXT VERTICAL BYTE
4043	7324	1D	.	.	DCR E	;ALL DONE?
4044	7325	F2	09	73	JP DAP020	;YES--GO THROUGH AGAIN
4045	7328	C3	C1	99	JMP GEXIT	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 114
=====
4047      732B      . . .      ;*****
4048      732B      . . .      ; SETORG--SET RELOCATABLE ORIGIN ABSOLUTE
4049      732B      . . .      ;*****
4050      732B      . . .      SETORG EQU $
4051      732B      3E 02 .      MVI A,2          ;GOT 2 VALUES?
4052      732D      CD 62 6D      CALL PRMCNT
4053      7330      C2 C1 99      JNZ GEXIT        ;NO, IGNORE
4054      7333      2A 89 90      LHLD PRMBUF      ;GET X COORD
4055      7336      EB . . .      XCHG
4056      7337      2A BB 90      LHLD PRMBUF+2    ;GET Y COORD
4057      733A      . . .      STORG1 EQU $
4058      733A      22 98 90      SHLD YORG        ;STORE Y
4059      733D      EB . . .      XCHG
4060      733E      22 9A 90      SHLD XORG        ;STORE X
4061      7341      C3 C1 99      JMP GEXIT
4062      7344      . . .      ;*****
4063      7344      . . .      ; PENORG--SET RELOC ORIGIN=CURRENT PEN POSITION
4064      7344      . . .      ;*****
4065      7344      . . .      PENORG EQU $
4066      7344      2A DE 90      LHLD XCURR
4067      7347      EB . . .      XCHG
4068      7348      2A DC 90      LHLD YCURR
4069      734B      C3 3A 73      JMP STORG1
4070      734E      . . .      ;*****
4071      734E      . . .      ; GCORG--SET RELOC ORIGIN=G CURSOR POSITION
4072      734E      . . .      ;*****
4073      734E      . . .      GCORG EQU $
4074      734E      2A CF 90      LHLD NEWGCX
4075      7351      EB . . .      XCHG
4076      7352      2A CD 90      LHLD NEWGCY
4077      7355      C3 3A 73      JMP STORG1
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 115
4079	7358	.	.	.	;*****	
4080	7358	.	.	.	; STATUS--ESC * S RECEIVED, SET UP FOR STATUS	
4081	7358	.	.	.	;*****	
4082	7358	.	.	.	STATUS EQU \$	
4083	7358	CD	82	B8	CALL APLTOF ;TURN AUTO PLOT OFF	
4084	7358	21	DC	61	LXI H,STATTB ;SET NEW RANGE TABLE	
4085	735E	C3	86	63	JMP SETRTB	
4086	7361	.	.	.	;*****	
4087	7361	.	.	.	; GSTAT--HAVE RECEIVED WHICH STATUS BLOCK USER	
4088	7361	.	.	.	; WANTS, GET READY TO SEND IT WHEN BLOCK XFER	
4089	7361	.	.	.	; TRIGGER RECEIVED	
4090	7361	.	.	.	;*****	
4091	7361	.	.	.	GSTAT EQU \$	
4092	7361	CD	D2	00	CALL ZCKRMT ;IN REMOTE MODE?	
4093	7364	CA	C1	99	JZ GEXIT ;NO, IGNORE STATUS REQUEST	
4094	7367	3E	0C	.	MVI A,MXSTAT ;MAXIMUM STATUS PARAMETER	
4095	7369	CD	43	6D	CALL GETPRM ;GET STATUS BLOCK PARAMETER	
4096	736C	C2	84	73	JNZ GSTO20 ;BAD, SEND ID STATUS BLOCK	
4097	736F	B7	.	.	ORA A ;VALUE = 0?	
4098	7370	CA	84	73	JZ GSTO20 ;YES, SEND ID STATUS BLOCK	
4099	7373	FE	04	.	CPI GCWBLK ;GRAPHICS CURSOR W/WAIT?	
4100	7375	CA	08	74	JZ GCWBGH ;YES, SET UP FOR IT	
4101	7378	.	.	.	GSTO10 EQU \$	
4102	7378	32	68	90	STA GSBLOK ;STORE STATUS BLOCK VALUE	
4103	7378	01	00	00	LXI B,0 ;SET BLOCK TRANSFER PENDING	
4104	737E	CD	D5	00	CALL ZSBXFR ;FLAGS	
4105	7381	C3	C1	99	JMP GEXIT	
4106	7384	.	.	.	GSTO20 EQU \$	
4107	7384	3E	01	.	MVI A,IDBLOK ;SEND ID IF BAD STATUS	
4108	7386	C3	78	73	JMP GSTO10 ;REQUEST	
4109	7389	.	.	.	;*****	
4110	7389	.	.	.	; STATGO--BLOCK TRANSFER TRIGGER RECEIVED, SEND	
4111	7389	.	.	.	; STATUS	
4112	7389	.	.	.	; ENTRY--DONT CARE	
4113	7389	.	.	.	; EXIT---ALL REGISTERS DESTROYED	
4114	7389	.	.	.	;*****	
4115	7389	.	.	.	STATGO EQU \$	
4116	7389	21	68	90	LXI H,GSBLOK ;GET VALUE OF STATUS BLOCK	
4117	738C	7E	.	.	MOV A,M	
4118	738D	F5	.	.	PUSH PSW ;SAVE IT	
4119	738E	36	00	.	MVI M,0 ;CLEAR XFER PENDING FLAG	
4120	7390	01	FF	FF	LXI B,-1 ;CLEAR MAIN CODE PENDING	
4121	7393	CD	D8	00	CALL ZCLBXF ;FLAGS	
4122	7396	3E	20	.	MVI A,SUPCHR ;TURN ECHO SUPPRESSION ON	
4123	7398	CD	4D	A2	CALL STTKFL	
4124	7398	F1	.	.	POP PSW ;RECALL STATUS BLOCK	
4125	739C	3D	.	.	DCR A ;WANT TO START AT 0	
4126	739D	21	B2	73	LXI H,STVECS ;ASSUME NOT TEK	
4127	73A0	.	.	.	; IF MSB SET, STATUS BLOCK IS FOR TEK	
4128	73A0	F2	A8	73	JP STG010 ;NOT TEK	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 116
=====
4129     73A3     21  CA   73                LXI  H,TKSTAT   ;BASE OF TEK VECTORS
4130     73A6     E6  03   .                ANI  3Q         ;DELETE SIGN BIT
4131     73A8     .   .   .          STG010 EQU  $
4132     73A8     87  .   .                ADD  A         ;2 * VALUE FOR INDEX TO TAB
4133     73A9     5F  .   .                MOV  E,A
4134     73AA     16  00   .                MVI  D,0       ;DE = INDEX TO STATUS TABLE
4135     73AC     19  .   .                DAD  D         ;FETCH ADDRESS OF STATUS
4136     73AD     5E  .   .                MOV  E,M       ;ROUTINE
4137     73AE     23  .   .                INX  H
4138     73AF     56  .   .                MOV  D,M
4139     73B0     EB  .   .                XCHG          ;JUMP TO PROPER ROUTINE
4140     73B1     E9  .   .                PCHL
4141     73B2     .   .   .                ;
4142     73B2     .   .   .          STVECS EQU  $
4143     73B2     D2  73   .                DW   IDGO      ;1--SEND ID
4144     73B4     DE  73   .                DW   PENG0    ;2--SEND PEN POSITION
4145     73B6     FE  73   .                DW   GCG0     ;3--GRAPHICS CURSOR LOC
4146     73B8     3E  74   .                DW   GCWGO    ;4--GRAPHICS CURSOR WITH WAI
4147     73BA     54  74   .                DW   DSPGO    ;5--DISPLAY SIZE
4148     73BC     84  74   .                DW   CAPGO    ;6--DEVICE CAPABILITIES
4149     73BE     AA  74   .                DW   TEXTGO   ;7--TEXT STATUS
4150     73C0     15  75   .                DW   ZOOMGO   ;8--ZOOM STATUS
4151     73C2     35  75   .                DW   ORGGO    ;9--RELOC. ORIGIN
4152     73C4     42  75   .                DW   PHYSGO   ;10--PHYSICAL STATUS
4153     73C6     6A  75   .                DW   AREAGO   ;11--AREA SHADING
4154     73C8     76  75   .                DW   DYNGO    ;12--DYNAMICS
4155     73CA     .   .   .                ;
4156     73CA     .   .   .                ;
4157     73CA     .   .   .          TKSTAT EQU  $
4158     73CA     86  6A   .                DW   TKACGO   ;1--ESC ENG IN A/N
4159     73CC     BF  6A   .                DW   TKCPGO   ;2--ESC ENG IN GRAPHICS
4160     73CE     42  6B   .                DW   TKGNGO   ;3--ESC SUB & KEY
4161     73D0     48  6B   .                DW   TKGCGU   ;4--ESC SUB ESC ENG
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 117
4163	73D2	.	.	*****	
4164	73D2	.	.	; IDGO--SEND TERMINAL IDENTIFICATION STRING	
4165	73D2	.	.	*****	
4166	73D2	.	.	IDGO EQU \$	
4167	73D2	21	D8 73	LXI H, IDSTR ; POINTER TO ID STRING	
4168	73D5	C3	B2 75	JMP SNDSTR ; SEND IT	
4169	73D8	.	.	;	
4170	73D8	.	.	IDSTR EQU \$	
4171	73D8	32	36 34	DB '2648A', 0	
4172	73DE	.	.	*****	
4173	73DE	.	.	; PENG0--SEND CURRENT PEN POSITION	
4174	73DE	.	.	*****	
4175	73DE	.	.	PENG0 EQU \$	
4176	73DE	2A	DC 90	LHLD YCURR ; Y COORD	
4177	73E1	EB	.	XCHG	
4178	73E2	2A	DE 90	LHLD XCURR ; X COORD	
4179	73E5	CD	9D 75	CALL SENDHD ; SEND THEM	
4180	73E8	CD	A8 75	CALL SNDCMA ; SEND A COMMA	
4181	73EB	.	.	; IF PEN IS UP, SEND 0, IF DOWN, SEND 1	
4182	73EB	0E	31 .	MVI C, ONE ; ASSUME ITS DOWN	
4183	73ED	3A	B2 90	LDA GFLGS1 ; IS MOVE BIT SET?	
4184	73F0	E6	01 .	ANI MOVE	
4185	73F2	CA	F7 73	JZ PNG010 ; NO, PEN REALLY IS DOWN	
4186	73F5	0E	30 .	MVI C, ZERO ; YES, PEN IS UP	
4187	73F7	.	.	PNG010 EQU \$	
4188	73F7	79	.	MOV A, C ; SEND PEN STATUS	
4189	73F8	CD	7C 00	CALL ZPUTDC	
4190	73FB	C3	C2 75	JMP SENDTM ; SEND TERMINATOR	
4191	73FE	.	.	*****	
4192	73FE	.	.	; GCG0--SEND GRAPHICS CURSOR POSITION	
4193	73FE	.	.	*****	
4194	73FE	.	.	GCG0 EQU \$	
4195	73FE	2A	CD 90	LHLD NEWGCY ; Y COORD	
4196	7401	EB	.	XCHG	
4197	7402	2A	CF 90	LHLD NEWGCX ; X COORD	
4198	7405	CD	9D 75	CALL SENDHD ; SEND THEM	
4199	7408	C3	C2 75	JMP SENDTM ; SEND TERMINATOR	
4200	740B	.	.	*****	
4201	740B	.	.	; GCWBGN--START GRAPHICS CURSOR WITH WAIT	
4202	740B	.	.	; DO NOT SEND X, Y UNTIL KEY IS HIT	
4203	740B	.	.	; SEND KEY CODE TOO	
4204	740B	.	.	; IF ESCAPE IS RECEIVED FROM ANYPLACE BUT KEYBOARD	
4205	740B	.	.	; THE STATUS REQUEST IS ABORTED	
4206	740B	.	.	; LOCAL 2-CHAR SEQUENCES, AS GENERATED BY FUNCTION	
4207	740B	.	.	; KEYS, ARE IGNORED	
4208	740B	.	.	*****	
4209	740B	.	.	GCWBGN EQU \$	
4210	740B	21	F0 61	LXI H, GCWTAB ; NEW RANGE TABLE	
4211	740E	C3	EA 6A	JMP STGIN1 ; DO THE REST AS WITH TEK	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 118
4213	7411	.	.	*****	
4214	7411	.	.	; GCWCHR--CHARACTER RECEIVED. IF LOCAL, SEND	
4215	7411	.	.	; IT AND CURSOR POSITION. IF REMOTE, IGNORE	
4216	7411	.	.	*****	
4217	7411	.	.	GCWCHR EQU \$	
4218	7411	CD	C3 00	CALL ZDCIO ;FROM KEYBOARD?	
4219	7414	C2	F0 00	JNZ ZCKCTL ;CHECK FOR BLOCK TRIGGER	
4220	7417	.	.	GCWCH1 EQU \$	
4221	7417	3E	04 .	MVI A,GCWBLK ;SET UP FOR XFER	
4222	7419	C3	1C 6B	JMP GINCH2	
4223	741C	.	.	*****	
4224	741C	.	.	; GCWESC--ESCAPE RECEIVED. IF REMOTE, ABORT	
4225	741C	.	.	; STATUS REQUEST. IF LOCAL AND CAME FROM	
4226	741C	.	.	; FUNCTION KEY, IGNORE IT. IF LOCAL AND SINGLE	
4227	741C	.	.	; KEY, USE IT AS THE KEYCODE	
4228	741C	.	.	*****	
4229	741C	.	.	GCWESC EQU \$	
4230	741C	CD	C3 00	CALL ZDCIO ;REMOTE?	
4231	741F	CA	2D 74	JZ GWE010 ;NO, INVESTIGATE FURTHER	
4232	7422	.	.	; REMOTE ESC RECEIVED, ABORT STATUS REQUEST AND	
4233	7422	.	.	; EXECUTE	
4234	7422	3E	30 .	MVI A,SUPCHR+GINMOD ;CLEAR ECHO SUPRESS,	
4235	7424	CD	53 A2	CALL CLTKFL ;GIN FLAGS	
4236	7427	CD	02 70	CALL TGCOF1 ;TURN THE CURSOR OFF	
4237	742A	C3	B7 00	JMP ZESCAP ;SET UP FOR ESCAPE	
4238	742D	.	.	GWE010 EQU \$	
4239	742D	.	.	; DID ESC COME FROM FUNCTION KEY?	
4240	742D	21	9C FF	LXI H,ZCHRIN ;FETCH ACTUAL KEYCODE	
4241	7430	7E	.	MOV A,M	
4242	7431	B7	.	ORA A ;2-CHAR SEQUENCE?	
4243	7432	F2	17 74	JP GCWCH1 ;NO, USE ESC AS INPUT CHAR	
4244	7435	E6	7F .	ANI 177Q ;YES, DISABLE 2ND CHAR	
4245	7437	77	.	MOV M,A	
4246	7438	.	.	; BACK TO NORMAL GCW TABLE	
4247	7438	21	F0 61	LXI H,GCWTAB ;WAIT FOR ANOTHER KEY	
4248	743B	C3	86 63	JMP SETRTB	
4249	743E	.	.	*****	
4250	743E	.	.	; GCWGO--BLOCK TRIGGER RECEIVED, SEND CHAR	
4251	743E	.	.	; AND CURSOR POSITION	
4252	743E	.	.	*****	
4253	743E	.	.	GCWGO EQU \$	
4254	743E	2A	D4 FB	LHLD YGINSV ;Y CURSOR COORD	
4255	7441	EB	.	XCHG	
4256	7442	2A	D6 FB	LHLD XGINSV ;X CURSOR COORD	
4257	7445	CD	9D 75	CALL SENDHD ;SEND THEM	
4258	7448	CD	A8 75	CALL SNDOMA ;SEND A COMMA	
4259	744B	3A	E2 FA	LDA GINCHR ;GET THE CHAR	
4260	744E	CD	80 75	CALL SENDA ;SEND IT	
4261	7451	C3	C2 75	JMP SENDTM ;SEND THE TERMINATOR	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 119
4263	7454	.	.	*****	
4264	7454	.	.	; DSPGO--SEND DISPLAY SIZE	
4265	7454	.	.	*****	
4266	7454	.	.	DSPGO EQU \$	
4267	7454	21	5A 74	LXI H,DSPY ;POINTER TO DISLAY STRING	
4268	7457	C3	B2 75	JMP SNDSTR ;SEND IT	
4269	745A	.	.	;	
4270	745A	.	.	DSPY EQU \$	
4271	745A	2B	30 30	DB '+00000,+00000,+00719,+00359,'	
4272	7476	30	30 30	DB '00003.,00003.',0	
4273	7484	.	.	*****	
4274	7484	.	.	; CAPGO--SEND DEVICE CAPABILITIES	
4275	7484	.	.	*****	
4276	7484	.	.	CAPGO EQU \$	
4277	7484	21	8A 74	LXI H,CAPSTR ;POINTER TO STRING	
4278	7487	C3	B2 75	JMP SNDSTR ;SEND IT	
4279	748A	.	.	;	
4280	748A	.	.	CAPSTR EQU \$	
4281	748A	33	2C 31	DB '3,1,0,0,1,0,0,1,'	
4282	749A	31	2C 31	DB '1,1,1,2,0,0,0,0',0	
4283	74AA	.	.	*****	
4284	74AA	.	.	; TEXTGO--SEND GRAPHICS TEXT STATUS	
4285	74AA	.	.	*****	
4286	74AA	.	.	TEXTGO EQU \$	
4287	74AA	.	.	; COMPUTE CELL SIZE BY MULTIPLYING BASIC	
4288	74AA	.	.	; SIZE (7 X BY 10 Y) BY TEXT SIZE	
4289	74AA	21	0A 00	LXI H,YCELL ;GET Y CELL SIZE	
4290	74AD	CD	2E 78	CALL MPYTSZ ;MULTIPLY BY TEXT SIZE	
4291	74B0	E5	.	PUSH H ;SAVE Y	
4292	74B1	21	07 00	LXI H,XCELL ;GET X CELL SIZE	
4293	74B4	CD	2E 78	CALL MPYTSZ	
4294	74B7	D1	.	POP D ;HL = X, DE = Y SIZE	
4295	74B8	CD	9D 75	CALL SENDHD ;SEND CELL SIZE	
4296	74BB	CD	A8 75	CALL SNDCMA ;SEND A COMMA	
4297	74BE	.	.	; NOW SEND TEXT ORIGIN	
4298	74BE	3A	D3 FB	LDA TXORG ;FETCH LABEL ORIGIN	
4299	74C1	C6	31 .	ADI 610 ;CONVERT TO ASCII NUMBER	
4300	74C3	CD	7C 00	CALL ZPUTDC ;SEND IT	
4301	74C6	.	.	; FETCH TEXT ANGLE IN DEGREES STRING FROM TABLE	
4302	74C6	3A	DB FB	LDA TANG ;GET THE ANGLE (0-3)	
4303	74C9	87	.	ADD A ;* 2	
4304	74CA	87	.	ADD A ;* 4	
4305	74CB	87	.	ADD A ;* 8	
4306	74CC	5F	.	MOV E,A ;USE AS INDEX	
4307	74CD	16	00 .	MVI D,0	
4308	74CF	21	E5 74	LXI H,SANGTB ;BASE OF TABLE	
4309	74D2	19	.	DAD D ;HL = PTR TO DEGREES STRING	
4310	74D3	CD	B8 75	CALL SNDST1 ;SEND ANGLE STRING	
4311	74D6	.	.	; NOW SEND SLANT IN DEGREES	
4312	74D6	21	05 75	LXI H,SLNTO ;ASSUME NO SLANT	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 120
4313	74D9	CD	42	76	CALL CKSLNT ;IS SLANT REALLY OFF?	
4314	74DC	CA	E2	74	JZ TXG010 ;YES	
4315	74DF	21	0D	75	LXI H,SLNT45 ;NO, SET FOR 45 DEGREES	
4316	74E2	.	.	.	TXG010 EQU \$	
4317	74E2	C3	B2	75	JMP SNDSTR ;SEND SLANT	
4318	74E5	.	.	.	;	
4319	74E5	.	.	.	SANGTB EQU \$	
4320	74E5	2C	30	30	DB ',00000.',0	
4321	74ED	2C	30	30	DB ',00090.',0	
4322	74F5	2C	30	30	DB ',00180.',0	
4323	74FD	2C	30	30	DB ',00270.',0	
4324	7505	.	.	.	;	
4325	7505	.	.	.	SLNT0 EQU \$	
4326	7505	2C	30	30	DB ',00000.',0	
4327	750D	.	.	.	SLNT45 EQU \$	
4328	750D	2C	30	30	DB ',00045.',0	
4329	7515	.	.	.	*****	
4330	7515	.	.	.	; ZOOMGO--SEND ZOOM STATUS	
4331	7515	.	.	.	*****	
4332	7515	.	.	.	ZOOMGO EQU \$	
4333	7515	.	.	.	; FIRST SEND SIZE	
4334	7515	3A	E1	FB	LDA MAG ;ZOOM MAGNIFICATION	
4335	7518	3C	.	.	INR A ;WANT 1-16, NOT 0-15	
4336	7519	CD	80	75	CALL SENDA ;SEND IT	
4337	751C	CD	AD	75	CALL SENDP ;SEND A DEC POINT	
4338	751F	CD	A8	75	CALL SNDCMA ;SEND A COMMA	
4339	7522	.	.	.	; SEND 0 IF ZOOM OFF, 1 IF ON	
4340	7522	0E	30	.	MVI C,ZERO ;ASSUME OFF	
4341	7524	3A	AE	90	LDA GFLG55 ;IS IT OFF?	
4342	7527	E6	02	.	ANI WANTZM	
4343	7529	CA	2E	75	JZ ZMG010 ;YES	
4344	752C	0E	31	.	MVI C,ONE ;NO	
4345	752E	.	.	.	ZMG010 EQU \$	
4346	752E	79	.	.	MOV A,C ;MOVE STATUS TO A	
4347	752F	CD	7C	00	CALL ZPUTDC ;SEND IT	
4348	7532	C3	C2	75	JMP SENDTM ;SEND THE TERMINATOR	
4349	7535	.	.	.	*****	
4350	7535	.	.	.	; ORGGO--SEND RELOCATABLE ORIGIN	
4351	7535	.	.	.	*****	
4352	7535	.	.	.	ORGGO EQU \$	
4353	7535	2A	98	90	LHLD YORG ;Y COORD	
4354	7538	EB	.	.	XCHG	
4355	7539	2A	9A	90	LHLD XORG ;X COORD	
4356	753C	CD	9D	75	CALL SENDHD ;SEND THEM	
4357	753F	C3	C2	75	JMP SENDTM ;SEND THE TERMINATOR	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 121
4359	7542	.	.	*****	
4360	7542	.	.	; PHYSGO--SEND PHYSICAL STATUS	
4361	7542	.	.	*****	
4362	7542	.	.	PHYSGO EQU \$	
4363	7542	.	.	; IF TERMINAL HAS BEEN RESET, SEND A 1 FOR	
4364	7542	.	.	; FIRST CHAR.	
4365	7542	0E	30	MVI C,ZERO ;ASSUME NO RESET	
4366	7544	3A	96 90	LDA GFLGS7	
4367	7547	E6	80	ANI RESET ;WAS IT RESET?	
4368	7549	CA	51 75	JZ PYG010 ;NO	
4369	754C	0E	31	MVI C,ONE ;YES	
4370	754E	CD	6D A2	CALL CLFLG7 ;CLEAR RESET FLAG	
4371	7551	.	.	PYG010 EQU \$	
4372	7551	79	.	MOV A,C ;MOVE STATUS TO A	
4373	7552	CD	7C 00	CALL ZPUTDC ;SEND THE FIRST ONE	
4374	7555	21	5B 75	LXI H,PHYSTR ;THE REST ARE ALL ZERO	
4375	7558	C3	B2 75	JMP SNDSTR	
4376	755B	.	.	;	
4377	755B	.	.	PHYSTR EQU \$	
4378	755B	2C	30 2C	DB ',0,0,0,0,0,0,0',0	
4379	756A	.	.	*****	
4380	756A	.	.	; AREAGO--SEND AREA SHADING CAPABILITY	
4381	756A	.	.	*****	
4382	756A	.	.	AREAGO EQU \$	
4383	756A	21	70 75	LXI H,ARSTR ;POINTER TO STRING	
4384	756D	C3	B2 75	JMP SNDSTR ;SEND IT	
4385	7570	.	.	;	
4386	7570	.	.	ARSTR EQU \$	
4387	7570	31	2C 38	DB '1,8,8',0	
4388	7576	.	.	*****	
4389	7576	.	.	; DYNGO--SEND DYNAMICS CAPABILITY	
4390	7576	.	.	*****	
4391	7576	.	.	DYNGO EQU \$	
4392	7576	21	7C 75	LXI H,DYNSTR ;POINTER TO STRING	
4393	7579	C3	B2 75	JMP SNDSTR ;SEND IT	
4394	757C	.	.	;	
4395	757C	.	.	DYNSTR EQU \$	
4396	757C	31	2C 31	DB '1,1',0	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 122
4398	7580	. . .	;*****	
4399	7580	. . .	; SENDA--CONVERT A REG TO DECIMAL FROM BINARY,	
4400	7580	. . .	; AND SEND TO DATACOM	
4401	7580	. . .	;*****	
4402	7580	. . .	SEDA EQU \$	
4403	7580	21 7C 00	LXI H,ZPUTDC ;ADDRESS OF OUTPUT ROUTINE	
4404	7583	C3 CC 00	JMP ZB2DA ;CONVERT AND SEND	
4405	7586	. . .	;*****	
4406	7586	. . .	; SENDHL--CONVERT HL TO DECIMAL AND SEND	
4407	7586	. . .	; INCLUDE SIGN	
4408	7586	. . .	;*****	
4409	7586	. . .	SENDHL EQU \$	
4410	7586	. . .	; GET ABSOLUTE VALUE	
4411	7586	7C . .	MOV A,H ;CHECK SIGN OF HL	
4412	7587	B7 . .	ORA A	
4413	7588	0E 2B .	MVI C,PLUS ;ASSUME POSITIVE	
4414	758A	F2 92 75	JP SHL010	
4415	758D	0E 2D .	MVI C,NEG ;REALLY MINUS	
4416	758F	CD 09 A3	CALL NEGATE ;CONVERT TO +	
4417	7592	. . .	SHL010 EQU \$	
4418	7592	79 . .	MOV A,C ;SIGN INTO A	
4419	7593	CD 7C 00	CALL ZPUTDC ;SEND SIGN	
4420	7596	EB . .	XCHG ;DE = VALUE	
4421	7597	21 7C 00	LXI H,ZPUTDC ;ADDRESS OF OUTPUT ROUTINE	
4422	759A	C3 CF 00	JMP ZB2DDE ;CONVERT AND SEND	
4423	759D	. . .	;*****	
4424	759D	. . .	; SENDHD--SEND VALUES IN HL AND DE, SEPARATED	
4425	759D	. . .	; BY COMMA	
4426	759D	. . .	;*****	
4427	759D	. . .	SENDHD EQU \$	
4428	759D	D5 . .	PUSH D ;SAVE D	
4429	759E	CD 86 75	CALL SENDHL ;SEND HL FIRST	
4430	75A1	CD A8 75	CALL SNDCMA ;SEND A COMMA	
4431	75A4	E1 . .	POP H ;NOW SEND D	
4432	75A5	C3 86 75	JMP SENDHL	
4433	75A8	. . .	;*****	
4434	75A8	. . .	; SNDCMA--SEND A COMMA	
4435	75A8	. . .	;*****	
4436	75A8	. . .	SNDCMA EQU \$	
4437	75A8	3E 2C .	MVI A,CMA	
4438	75AA	C3 7C 00	JMP ZPUTDC	
4439	75AD	. . .	;*****	
4440	75AD	. . .	; SENDP--SEND A DECIMAL POINT	
4441	75AD	. . .	;*****	
4442	75AD	. . .	SENDP EQU \$	
4443	75AD	3E 2E .	MVI A,POINT	
4444	75AF	C3 7C 00	JMP ZPUTDC	
4445	75B2	. . .	;*****	
4446	75B2	. . .	; SNDSTR--SEND A STRING AND TERMINATOR	
4447	75B2	. . .	; ENTRY HL = POINTER TO STRING	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 123
=====
4448      75B2      . . .      ; STRING TERMINATED BY NULL BYTE
4449      75B2      . . .      ;*****
4450      75B2      . . .      SNDSTR EQU $
4451      75B2      CD B8 75      CALL SNDST1      ;SEND THE STRING
4452      75B5      C3 C2 75      JMP SENDTM      ;SEND THE TERMINATOR
4453      75B8      . . .      SNDST1 EQU $
4454      75B8      7E . .      MOV A,M      ;FETCH BYTE
4455      75B9      B7 . .      ORA A      ;END OF STRING?
4456      75BA      C8 . .      RZ      ;YES, DONE
4457      75BB      CD 7C 00      CALL ZPUTDC      ;NO, SEND CHARACTER
4458      75BE      23 . .      INX H      ;GET NEXT CHAR
4459      75BF      C3 B8 75      JMP SNDST1
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 124
=====
4461      75C2      . . .      ;*****
4462      75C2      . . .      ; SENDTM--SEND TERMINATING CHAR, THEN SET UP
4463      75C2      . . .      ; TO IGNORE CR/LF (IF ECHOED)
4464      75C2      . . .      ;*****
4465      75C2      . . .      SENDTM EQU $
4466      75C2      CD DB 00      CALL ZSDTRM      ;SEND THE TERMINATOR
4467      75C5      . . .      ; FALL INTO NOCRLF
4468      75C5      . . .      ;*****
4469      75C5      . . .      ; NOCRLF--IGNORE CR/LF IF NOT FROM DATACOM
4470      75C5      . . .      ; CALLED AFTER STATUS XFER OR AUTO PLOT SEQUENCE
4471      75C5      . . .      ;*****
4472      75C5      . . .      NOCRLF EQU $
4473      75C5      21 A6 62      LXI H,CRLFTB      ;SET NEW RANGE TABLE
4474      75C8      CD 86 63      CALL SETRTB
4475      75CB      . . .      NODC3 EQU $      ;(IGNORE DC3)
4476      75CB      21 D1 FF      LXI H,ZESCFG      ;SEQ 2 CHAR COUNTER GOING
4477      75CE      36 02 .      MVI M,2
4478      75D0      C9 . .      RET
4479      75D1      . . .      ;*****
4480      75D1      . . .      ; NOCR--CARRIAGE RETURN RECEIVED
4481      75D1      . . .      ;*****
4482      75D1      . . .      NOCR EQU $
4483      75D1      CD C3 00      CALL ZDCIO      ;FROM KEYBOARD?
4484      75D4      . . .      ; IF EXECUTED, RANGE TABLES ARE RESET IN CHINT
4485      75D4      CA C0 00      JZ ZCRRET      ;YES, PROCESS THE CR
4486      75D7      C3 CB 75      JMP NODC3      ;NO, IGNORE
4487      75DA      . . .      ;*****
4488      75DA      . . .      ; NOLF--LINE FEED RECEIVED
4489      75DA      . . .      ;*****
4490      75DA      . . .      NOLF EQU $
4491      75DA      CD C3 00      CALL ZDCIO      ;FROM KEYBOARD?
4492      75DD      CA 8B 00      JZ ZLNFD      ;YES, EXECUTE
4493      75E0      C9 . .      RET      ;NO, IGNORE
4494      75E1      . . .      ; RANGE TABLES ARE RESET IN CHINT
4495      75E1      . . .      ;*****
4496      75E1      . . .      ; LBLEND--ENDED LABEL MODE WITH CR OR LF, BUT
4497      75E1      . . .      ; NEXT CHAR WASNT LF OR CR. REPEAT CHAR WITH
4498      75E1      . . .      ; NORMAL RANGE TABLE
4499      75E1      . . .      ;*****
4500      75E1      . . .      LBLEND EQU $
4501      75E1      CD 9D 6C      CALL LBLOFF      ;TURN LABEL MODE OFF
4502      75E4      . . .      ; FALL INTO REPEAT CHAR ROUTINE
4503      75E4      . . .      ;*****
4504      75E4      . . .      ; CRLFON--SOME OTHER CHARACTER RECEIVED
4505      75E4      . . .      ; LOAD NORMAL RANGE TABLE AND REPEAT
4506      75E4      . . .      ;*****
4507      75E4      . . .      CRLFON EQU $
4508      75E4      CD 4F 00      CALL ZESCND      ;RESTORE RANGE TABLES
4509      75E7      21 88 FF      LXI H,ZCHAR      ;FETCH THE CHAR
4510      75EA      4E . .      MOV C,M      ;PUT INTO C REG

```



```
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 125
4511	75EB	CD	82	00	CALL ZCHINT	;EXECUTE IT
4512	75EE	C3	CB	75	JMP NODC3	;RESET 2 CHAR COUNTER

```
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 126
4514	75F1	.	.	*****	
4515	75F1	.	.	; GRAPHICS TEXT ROUTINES	
4516	75F1	.	.	*****	
4517	75F1	.	.	;	
4518	75F1	.	.	*****	
4519	75F1	.	.	; GTXON--TURN GRAPHICS TEXT MODE ON	
4520	75F1	.	.	*****	
4521	75F1	.	.	GTXON EQU \$	
4522	75F1	CD	F7 75	CALL GTXON1	
4523	75F4	C3	C1 99	JMP GEXIT	
4524	75F7	.	.	GTXON1 EQU \$;(INTERNAL ENTRY)	
4525	75F7	.	.	; DONT TURN GTEXT ON IF AUTO PLOT ON	
4526	75F7	CD	F4 88	CALL CHEKAP ;IS AUTO PLOT ON?	
4527	75FA	C0	.	RNZ ;YES, LEAVE TEXT OFF	
4528	75FB	3E	02 .	MVI A,GTEXT ;TURN GRAPHICS TEXT ON	
4529	75FD	CD	5A A2	CALL STFLG6	
4530	7600	CD	9D 6C	CALL LBLOFF	
4531	7603	.	.	; IF TURN ON TEXT MODE WHILE CURSOR IS ON, MOVE	
4532	7603	.	.	; CURRENT POINT TO CURSOR	
4533	7603	3A	80 90	LDA GFLGS3 ;IS CURSOR ON?	
4534	7606	E6	80 .	ANI WANTGC	
4535	7608	C8	.	RZ ;NO,DONE	
4536	7609	2A	CF 90	LHLD NEWGCX ;UPDATE CURRENT POINT	
4537	760C	22	DA 90	SHLD XNEW	
4538	760F	2A	CD 90	LHLD NEWGCY	
4539	7612	22	DB 90	SHLD YNEW	
4540	7615	C3	32 98	JMP CPUPDA	
4541	7618	.	.	*****	
4542	7618	.	.	; GTXOF--TURN GRAPHICS TEXT MODE OFF	
4543	7618	.	.	*****	
4544	7618	.	.	GTXOF EQU \$	
4545	7618	CD	1E 76	CALL GTXOF1	
4546	761B	C3	C1 99	JMP GEXIT	
4547	761E	.	.	GTXOF1 EQU \$;(INTERNAL ENTRY)	
4548	761E	CD	BD 00	CALL ZCRADV ;CLEAR CURSOR ADVANCE FLAG	
4549	7621	3E	82 .	MVI A,GTEXT+LABEL ;GTEXT OFF	
4550	7623	C3	60 A2	JMP CLFLG6	
4551	7626	.	.	*****	
4552	7626	.	.	; SLNTON--TURN TEXT SLANT ON	
4553	7626	.	.	*****	
4554	7626	.	.	SLNTON EQU \$	
4555	7626	CD	5C 69	CALL CKSCLO ;DONT CHANGE IF IN SCALED TE	
4556	7629	C2	C1 99	JNZ GEXIT	
4557	762C	3E	01 .	MVI A,SLANT	
4558	762E	CD	5A A2	CALL STFLG6 ;TURN ON SLANT	
4559	7631	.	.	SLNT1 EQU \$	
4560	7631	3A	DB FB	LDA TANG ;RECOMPUTE PARAMETERS USING	
4561	7634	CD	48 76	CALL ANGLE ;PROPER SLANT VALUE	
4562	7637	C3	C1 99	JMP GEXIT	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 127
=====
4564     763A      . . .      ;*****
4565     763A      . . .      ; SLNTOF--TURN SLANT OFF
4566     763A      . . .      ;*****
4567     763A      . . .      SLNTOF EQU $
4568     763A      3E 01 .      MVI A,SLANT
4569     763C      CD 60 A2     CALL CLFLG6 ;TURN OFF SLANT
4570     763F      C3 31 76     JMP SLNT1 ;RECOMPUTE CHAR. PARAMS
4571     7642      . . .      ;*****
4572     7642      . . .      ; CKSLNT--SEE IF SLANTED TEXT IS ON
4573     7642      . . .      ; EXIT NZ => SLANT IS ON, A DESTROYED
4574     7642      . . .      ;*****
4575     7642      . . .      CKSLNT EQU $
4576     7642      3A 97 90     LDA GFLGS6
4577     7645      E6 01 .      ANI SLANT
4578     7647      C9 . .      RET
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 128
4580	7648	.	.	*****	
4581	7648	.	.	; ANGLE--COMPUTE VECTOR PARAMETERS USING SPECIFIED	
4582	7648	.	.	; ANGLE (0-3). MUST RECOMPUTE THESE PARAMETERS	
4583	7648	.	.	; WHENEVER, ANGLE, SIZE OR SLANT IS CHANGED	
4584	7648	.	.	; ENTRY A = ANGLE	
4585	7648	.	.	*****	
4586	7648	.	.	ANGLE EQU \$	
4587	7648	32	DB FB	STA TANG ;STORE NEW ANGLE	
4588	7648	.	.	; COMPUTE INDEX TO PARAMETER TABLE--27 ENTRIES	
4589	7648	.	.	; PER QUADRENT. MULTIPLY ANGLE (0-3) BY 27	
4590	7648	5F	.	MOV E,A	
4591	764C	16	00 .	MVI D,0 ;DE = ANGLE	
4592	764E	68	.	MOV L,E	
4593	764F	62	.	MOV H,D ;HL = ANGLE	
4594	7650	29	.	DAD H ;2 * ANGLE	
4595	7651	29	.	DAD H ;4 * ANGLE	
4596	7652	29	.	DAD H ;8 * ANGLE	
4597	7653	19	.	DAD D ;9 * ANGLE	
4598	7654	5D	.	MOV E,L ;DE = 9 * ANGLE	
4599	7655	54	.	MOV D,H	
4600	7656	19	.	DAD D ;18 * ANGLE	
4601	7657	19	.	DAD D ;27 * ANGLE	
4602	7658	.	.	; COMPUTE POINTER TO FIRST ENTRY IN TABLE	
4603	7658	11	70 77	LXI D,ANGTAB ;BASE OF TABLE	
4604	765B	19	.	DAD D ;HL = POINTER TO 1ST PARM	
4605	765C	.	.	;TRANSFER FIRST BLOCK FROM TABLE TO RAM	
4606	765C	11	95 90	LXI D,CFM1+2 ;FIRST DESTINATION BYTE	
4607	765F	01	13 00	LXI B,19 ;NO. OF BYTES	
4608	7662	CD	28 AB	CALL XFER ;XFER IN REVERSE ORDER	
4609	7665	.	.	; IF SLANT IS ON, MUST USE SLANTED CHAR FILL AND	
4610	7665	.	.	; SIZE--LOAD THEM OVER THE FIRST 8 BYTES.	
4611	7665	CD	42 76	CALL CKSLNT ;IS SLANT ON?	
4612	7668	CA	74 76	JZ ANG010 ;NO, LEAVE AS IS	
4613	7668	.	.	; HL = POINTER TO START OF SLANTED PARAMETERS	
4614	7668	11	92 90	LXI D,CFXINC+1 ;FIRST SLANTED BYTE	
4615	766E	01	08 00	LXI B,8 ;NO. OF SLANTED PARAMS.	
4616	7671	CD	28 AB	CALL XFER ;WIPE OUT WHATS ALREADY THER	
4617	7674	.	.	; CHARACTER SIZE, CHARACTER SPACING, AND LINE FEED	
4618	7674	.	.	; SPACING MUST BE MULTIPLIED BY SIZE.	
4619	7674	.	.	ANG010 EQU \$	
4620	7674	21	8D 90	LXI H,XCHSIZ ;FIRST TO BE MULTIPLIED	
4621	7677	3E	06 .	MVI A,6 ;NO OF DATA BYTE PAIRS	
4622	7679	CD	17 78	CALL MPYALL ;REDO THEM	
4623	767C	.	.	; VECTOR LENGTH ALSO DEPENDS ON SIZE	
4624	767C	21	F9 FF	LXI H,-7 ;SMALLEST SIZE	
4625	767F	CD	2E 78	CALL MPYTSZ	
4626	7682	22	D8 FB	SHLD CHLEN ;STORE VECTOR LENGTH	
4627	7685	.	.	; MUST ADD COMPENSATING FACTOR TO SIZE FOR SLANTED	
4628	7685	.	.	; CHAR SO BOUNDS CHECK IS OK	
4629	7685	CD	42 76	CALL CKSLNT ;IS SLANT ON?	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 129
=====
4630      7688      CA B2 76      JZ   ANG015      ;NO
4631      768B      .   .   .      ; ADD SLANTED FUDGE FACTOR
4632      768B      3A DA FB      LDA  TXMAG      ;USE SIZE AS FUDGE FACTOR
4633      768E      6F   .   .      MOV  L,A
4634      768F      26 00 .      MVI  H,0      ;HL = CORRECTION FACTOR
4635      7691      .   .   .      ; IF ANGLE IS 2 OR 3, SUBTRACT THE CORRECTION
4636      7691      3A DB FB      LDA  TANG      ;FETCH ANGLE
4637      7694      4F   .   .      MOV  C,A      ;SAVE IT
4638      7695      E6 02 .      ANI  20
4639      7697      C4 09 A3      CNZ  NEGATE     ;NEGATE IF ANGLE = 2 OR 3
4640      769A      EB   .   .      XCHG          ;DE = CORRECTION
4641      769B      .   .   .      ; IF ANGLE IS 0 OR 2, ADD TO X
4642      769B      .   .   .      ; IF ANGLE IS 1 OR 3, ADD TO Y
4643      769B      3E 01 .      MVI  A,10     ;TEST LSBIT
4644      769D      A1   .   .      ANA  C
4645      769E      C2 AB 76      JNZ  ANG013    ;ADD TO Y
4646      76A1      .   .   .      ; ADD HL TO X CHAR SIZE
4647      76A1      2A 8D 90      LHLD XCHSIZ
4648      76A4      19   .   .      DAD  D
4649      76A5      22 8D 90      SHLD XCHSIZ
4650      76A8      C3 B2 76      JMP  ANG015
4651      76AB      .   .   .      ;ADD FUDGE TO Y CHAR SIZE
4652      76AB      .   .   .      ANG013 EQU $
4653      76AB      2A 8B 90      LHLD YCHSIZ
4654      76AE      19   .   .      DAD  D
4655      76AF      22 8B 90      SHLD YCHSIZ
4656      76B2      .   .   .      ANG015 EQU $
4657      76B2      .   .   .      ; BOTH SLANTED AND UNSLANTED SIZES ARE STILL OFF
4658      76B2      .   .   .      ; BY ONE. ADD 1 IF SIZE IS -, SUBTRACT IF SIZE IS
4659      76B2      .   .   .      ; +
4660      76B2      2A 8D 90      LHLD XCHSIZ   ;FIX UP X
4661      76B5      CD 59 77      CALL ANG050
4662      76B8      22 8D 90      SHLD XCHSIZ
4663      76BB      2A 8B 90      LHLD YCHSIZ   ;FIX UP Y
4664      76BE      CD 59 77      CALL ANG050
4665      76C1      22 8B 90      SHLD YCHSIZ
4666      76C4      .   .   .      ; MUST COMPUTE ADJUSTMENT FOR MIDDLE OR TOP OF
4667      76C4      .   .   .      ; TEXT. USE LF INC FOR TOP, OR (LFINC+-TXMAG)/2
4668      76C4      .   .   .      ; FOR MIDDLE.
4669      76C4      3A 97 90      LDA  GFLGS6    ;SET FOR BOT OF CHAR?
4670      76C7      4F   .   .      MOV  C,A
4671      76C8      E6 60 .      ANI  TOPCH+MIDCH
4672      76CA      CA 4F 77      JZ   ANG040    ;YES, USE 0 ADJUSTMENT
4673      76CD      3E 40 .      MVI  A, TOPCH  ;SET FOR TOP OF CHAR?
4674      76CF      A1   .   .      ANA  C
4675      76D0      CA 00 77      JZ   ANG020    ;NO, COMPUTE FOR MID OF CH
4676      76D3      .   .   .      ; TO ADJUST FOR TOP OF CHAR, USE LF INCREMENT
4677      76D3      2A 85 90      LHLD XLFINC
4678      76D6      22 77 90      SHLD XCHADJ
4679      76D9      2A 83 90      LHLD YLFINC
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 130
4680	76DC	22	75	90	SHLD YCHADJ
4681	76DF	.	.	.	; IF SLANTED, ADD AN ADDITIONAL ADJUSTMENT TO
4682	76DF	.	.	.	; ACCOUNT FOR SHIFTED CENTER OF CHAR
4683	76DF	.	.	.	; (SO LONG WILL BE CORRECT)
4684	76DF	CD	42	76	CALL CKSLNT ;IS SLANT ON?
4685	76E2	C8	.	.	RZ ;NO, DONE
4686	76E3	.	.	.	; ADD ADJUSTMENT
4687	76E3	.	.	.	; ADD YLFINC-1 TO X, AND XLF TO Y (ANGLES 0,2)
4688	76E3	.	.	.	; ADD -1*XLF-1 TO Y, AND YLF TO X (ANGLES 1,3)
4689	76E3	2A	83	90	LHLD YLFINC ;UPDATE X
4690	76E6	CD	64	77	CALL ANG060 ;DECREMENT (& NEGATE)
4691	76E9	EB	.	.	XCHG
4692	76EA	2A	77	90	LHLD XCHADJ ;X ADJUSTMENT
4693	76E0	19	.	.	DAD D
4694	76EE	22	77	90	SHLD XCHADJ
4695	76F1	2A	85	90	LHLD XLFINC ;UPDATE Y
4696	76F4	CD	64	77	CALL ANG060 ;DECREMENT (& NEGATE)
4697	76F7	EB	.	.	XCHG
4698	76F8	2A	75	90	LHLD YCHADJ
4699	76FB	19	.	.	DAD D
4700	76FC	22	75	90	SHLD YCHADJ
4701	76FF	C9	.	.	RET
4702	7700	.	.	.	; ADJUST FOR MIDDLE OF CHAR
4703	7700	.	.	.	ANG020 EQU \$
4704	7700	2A	85	90	LHLD XLFINC ;ADD TEXT SIZE, DIVIDE
4705	7703	CD	39	77	CALL ANG030
4706	7706	22	77	90	SHLD XCHADJ
4707	7709	2A	83	90	LHLD YLFINC
4708	770C	CD	39	77	CALL ANG030
4709	770F	22	75	90	SHLD YCHADJ
4710	7712	.	.	.	; IF SLANTED, ADD AN ADDITIONAL ADJUSTMENT TO
4711	7712	.	.	.	; ACCOUNT FOR SHIFTED CENTER OF CHAR]
4712	7712	.	.	.	; (SO LONG WILL BE CORRECT)
4713	7712	CD	42	76	CALL CKSLNT ;IS SLANT ON?
4714	7715	C8	.	.	RZ ;NO,DONE
4715	7716	.	.	.	; ADD YLF/2 TO X, AND XLF/2 TO Y (ANGLES 0,2)
4716	7716	.	.	.	; ADD -1*YLF/2 TO X, AND XLF/2 TO Y (ANGLES 1,3)
4717	7716	2A	83	90	LHLD YLFINC ;UPDATE X
4718	7719	CD	67	77	CALL ANG065 ;NEGATE IF NECESSARY
4719	771C	CD	39	77	CALL ANG030 ;GET LF-TEXT SIZE FACTOR
4720	771F	EB	.	.	XCHG
4721	7720	2A	77	90	LHLD XCHADJ ;FETCH X ADJUSTMENT
4722	7723	19	.	.	DAD D
4723	7724	22	77	90	SHLD XCHADJ
4724	7727	2A	85	90	LHLD XLFINC ;NOW DO Y
4725	772A	CD	67	77	CALL ANG065 ;NEGATE IF NECESSARY
4726	772D	CD	39	77	CALL ANG030 ;GET LF-TEXT SIZE FACTOR
4727	7730	EB	.	.	XCHG
4728	7731	2A	75	90	LHLD YCHADJ
4729	7734	19	.	.	DAD D

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 131
4730	7735	22	75	90	SHLD YCHADJ	
4731	7738	C9	.	.	RET	
4732	7739	.	.	.	ANG030 EQU \$	
4733	7739	.	.	.	; DONT CHANGE HL IF IT IS 0	
4734	7739	7D	.	.	MOV A,L	
4735	773A	B4	.	.	ORA H	
4736	773B	C8	.	.	RZ	
4737	773C	.	.	.	; DIVIDE LFINC BY 2	
4738	773C	CD	1A	A3	CALL DIVHL1 ;HL = HL/2	
4739	773F	EB	.	.	XCHG	
4740	7740	3A	DA	FB	LDA TXMAG ;DIVIDE TX SIZE BY 2	
4741	7743	B7	.	.	ORA A ;(CLEAR CARRY)	
4742	7744	1F	.	.	RAR	
4743	7745	6F	.	.	MOV L,A ;SET HL = TEXT SIZE	
4744	7746	26	00	.	MVI H,0	
4745	7748	7A	.	.	MOV A,D ;SET HL SAME SIGN AS LFINC	
4746	7749	B7	.	.	ORA A ;IS IT NEGATIVE?	
4747	774A	FC	09	A3	CM NEGATE ;YES, SET HL - TOO	
4748	774D	19	.	.	DAD D ;HL = LFINC/2 + TXSIZ/2	
4749	774E	C9	.	.	RET	
4750	774F	.	.	.	ANG040 EQU \$	
4751	774F	.	.	.	; SET FOR BOT OF CHAR, USE 0 INCREMENT	
4752	774F	21	00	00	LXI H,0	
4753	7752	22	77	90	SHLD XCHADJ	
4754	7755	22	75	90	SHLD YCHADJ	
4755	7758	C9	.	.	RET	
4756	7759	.	.	.	ANG050 EQU \$	
4757	7759	.	.	.	; ADD 1 TO HL IF -, SUBTRACT 1 IF 0	
4758	7759	.	.	.	; DO NOTHING IF 0	
4759	7759	7C	.	.	MOV A,H ;CHECK SIGN	
4760	775A	B7	.	.	ORA A	
4761	775B	FA	62	77	JM ANG055 ;ITS -	
4762	775E	B5	.	.	ORA L ;TEST FOR 0	
4763	775F	C8	.	.	RZ ;IT IS, DONE	
4764	7760	2B	.	.	DCX H ;SUBTRACT 1	
4765	7761	C9	.	.	RET	
4766	7762	.	.	.	ANG055 EQU \$	
4767	7762	23	.	.	INX H ;ADD 1	
4768	7763	C9	.	.	RET	
4769	7764	.	.	.	ANG060 EQU \$	
4770	7764	CD	59	77	CALL ANG050 ;ADD/SUB 1 TO HL	
4771	7767	.	.	.	ANG065 EQU \$	
4772	7767	.	.	.	; IF ANGLE IS 1 OR 3, NEGATE HL	
4773	7767	3A	DB	FB	LDA TANG ;FETCH ANGLE	
4774	776A	E6	01	.	ANI 1Q ;TEST LSB	
4775	776C	C8	.	.	RZ ;NOT 1 OR 3, EXIT	
4776	776D	C3	09	A3	JMP NEGATE ;NEGATE HL	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 132
=====
4778      7770      . . .      ;*****
4779      7770      . . .      ; TABLE OF TEXT PARAMETERS BY QUADRANT
4780      7770      . . .      ; STORED MSBYTE, THEN LSBYTE
4781      7770      . . .      ;*****
4782      7770      . . .      ANGTAB EQU $
4783      7770      . . .      ; 1ST QUADRANT
4784      7770      00 00 01      DB 0,0,1      ;M1 = 1
4785      7773      00 00 .      DB 0,0      ;X FILL = 0
4786      7775      FF D3 .      DB 377Q,323Q ;Y FILL INC = -45
4787      7777      00 07 .      DB 0,7      ;X CHAR SIZE = 7
4788      7779      00 0A .      DB 0,10     ;Y CHAR SIZE = 10
4789      777B      00 07 .      DB 0,7      ;X CHAR SPACING = 7
4790      777D      00 00 .      DB 0,0      ;Y CHAR SPACING = 0
4791      777F      00 00 .      DB 0,0      ;X LF = 0
4792      7781      FF F6 .      DB 377Q,366Q ;YLF = -10
4793      7783      00 01 .      DB 0,1      ;X FILL SLANTED = 1
4794      7785      FF D3 .      DB 377Q,323Q ;Y FILL SLANTED = -45
4795      7787      00 10 .      DB 0,16     ;X SLANTED SIZE = 16
4796      7789      00 0A .      DB 0,10     ;Y SLANTED SIZE = 10
4797      778B      . . .      ; 2ND QUADRANT
4798      778B      FF FD 30     DB 377Q,375Q,60Q ;M1 = -720
4799      778E      FF FF .      DB 377Q,377Q ;X FILL = -1
4800      7790      00 00 .      DB 0,0      ;Y FILL INC = 0
4801      7792      FF F6 .      DB 377Q,566Q ;X CHAR SIZE = -10
4802      7794      00 07 .      DB 0,7      ;X CHAR SIZE = 7
4803      7796      00 00 .      DB 0,0      ;X CHAR SPACING = 0
4804      7798      00 07 .      DB 0,7      ;Y CHAR SPACING = 7
4805      779A      00 0A .      DB 0,10     ;X LF = 10
4806      779C      00 00 .      DB 0,0      ;Y LF = 0
4807      779E      FF FF .      DB 377Q,377Q ;X SLANT INC = -1
4808      77A0      FF D3 .      DB 377Q,323Q ;Y SLANT INC = -45
4809      77A2      FF F6 .      DB 377Q,366Q ;X SLANTED SIZE = -10
4810      77A4      00 10 .      DB 0,16     ;Y SLANTED SIZE = 16
4811      77A6      . . .      ; 3RD QUADRANT
4812      77A6      FF FF FF     DB 377Q,377Q,377Q ;M1 = -1
4813      77A9      00 00 .      DB 0,0      ;X FILL = 0
4814      77AB      00 2D .      DB 0,45     ;Y FILL = 45
4815      77AD      FF F9 .      DB 377Q,371Q ;X CHAR SIZE = -7
4816      77AF      FF F6 .      DB 377Q,366Q ;Y CHAR SIZE = -10
4817      77B1      FF F9 .      DB 377Q,371Q ;X CHAR SPACING = -7
4818      77B3      00 00 .      DB 0,0      ;Y CHAR SPACING = 0
4819      77B5      00 00 .      DB 0,0      ;X LF = 0
4820      77B7      00 0A .      DB 0,10     ;Y LF = 10
4821      77B9      FF FF .      DB 377Q,377Q ;X SLANTED FILL = -1
4822      77BB      00 2D .      DB 0,45     ;Y SLANTED FILL = 45
4823      77BD      FF F0 .      DB 377Q,360Q ;X SLANTED SIZE = -16
4824      77BF      FF F6 .      DB 377Q,366Q ;Y SLANTED SIZE = -10
4825      77C1      . . .      ; 4TH QUADRANT
4826      77C1      00 02 D0     DB 0,2Q,320Q ;M1 = 720
4827      77C4      00 01 .      DB 0,1      ;X FILL = 1
=====

```



```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 133
=====
```

4828	77C6	00 00 .	DB 0,0 ;Y FILL = 0
4829	77C8	00 0A .	DB 0,10 ;X CHAR SIZE = 10
4830	77CA	FF F9 .	DB 377Q,371Q ;Y CHAR SIZE = -7
4831	77CC	00 00 .	DB 0,0 ;X CHAR SPACING = 0
4832	77CE	FF F9 .	DB 377Q,371Q ;Y CHAR SPACING = -7
4833	77D0	FF F6 .	DB 377Q,366Q ;X LF = -10
4834	77D2	00 00 .	DB 0,0 ;Y LF = 0
4835	77D4	00 01 .	DB 0,1 ;X SLANTED FILL = 1
4836	77D6	00 2D .	DB 0,45 ;Y SLANTED FILL = 45
4837	77D8	00 0A .	DB 0,10 ;X SLANTED SIZE = 10
4838	77DA	FF F0 .	DB 377Q,360Q ;Y SLANTED SIZE = -16

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 134
4840	77DC	.	.	;*****	
4841	77DC	.	.	; TXSIZE--CHANGE TEXT SIZE (1-8)	
4842	77DC	.	.	;*****	
4843	77DC	.	.	TXSIZE EQU \$	
4844	77DC	CD	5C 69	CALL CKSCLD ;DONT CHANGE IF SCALED TEK 0	
4845	77DF	C2	C1 99	JNZ GEXIT	
4846	77E2	3E	08 .	MVI A,8 ;MAX VALUE	
4847	77E4	CD	43 6D	CALL GETPRM ;GET PARAMETER	
4848	77E7	C2	C1 99	JNZ GEXIT ;IGNORE IF BAD	
4849	77EA	3D	. .	DCR A ;WANT 0-7, NOT 1-8	
4850	77EB	F4	F1 77	CP TXSIZ1 ;IGNORE IF IT WAS 0	
4851	77EE	C3	C1 99	JMP GEXIT	
4852	77F1	.	.	TXSIZ1 EQU \$	
4853	77F1	32	DA FB	STA TXMAG ;STORE MAGNIFICATION	
4854	77F4	3A	DB FB	LDA TANG ;RECOMPUTE PARMS WITH NEW SI	
4855	77F7	C3	48 76	JMP ANGLE	
4856	77FA	.	.	;*****	
4857	77FA	.	.	; RUM BREAK 3	
4858	77FA	.	.	ORG ZBRK2+4000Q	
4859	7800	.	.	ZBRK3 EQU \$	
4860	7800	54	. .	DB VERSN	
4861	7801	78	. .	DB ZBRK3/256	
4862	7802	.	.	;*****	
4863	7802	.	.	;*****	
4864	7802	.	.	; TXANGL--SET TEXT ANGLE (1-4)	
4865	7802	.	.	;*****	
4866	7802	.	.	TXANGL EQU \$	
4867	7802	CD	5C 69	CALL CKSCLD ;DONT CHANGE IF SCALED TEK 0	
4868	7805	C2	C1 99	JNZ GEXIT	
4869	7808	3E	04 .	MVI A,4 ;MAX VALUE	
4870	780A	CD	43 6D	CALL GETPRM ;GET PARAMETER	
4871	780D	C2	C1 99	JNZ GEXIT ;IGNORE IF BAD	
4872	7810	3D	. .	DCR A ;WANT 0-3, NOT 1-4	
4873	7811	F4	48 76	CP ANGLE ;IGNORE IF IT WAS 0	
4874	7814	C3	C1 99	JMP GEXIT	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 135
4876	7817	.	.	;*****	
4877	7817	.	.	; MPYALL--MULTIPLY TEXT PARAMETERS BY TEXT	
4878	7817	.	.	; SIZE.	
4879	7817	.	.	; ENTRY HL = POINTER TO FIRST PARAMETER	
4880	7817	.	.	; A = NUMBER OF PARAMETERS	
4881	7817	.	.	;*****	
4882	7817	.	.	MPYALL EQU \$	
4883	7817	3D	.	DCR A ;UPDATE COUNT	
4884	7818	F8	.	RM ;RETURN WHEN ALL DONE	
4885	7819	F5	.	PUSH PSW ;SAVE COUNT	
4886	781A	E5	.	PUSH H ;SAVE PARAMETER POINTER	
4887	781B	5E	.	MOV E,M ;FETCH LSBYTE OF PARAMETER	
4888	781C	23	.	INX H	
4889	781D	56	.	MOV D,M ;FETCH MSBYTE	
4890	781E	EB	.	XCHG ;HL = PARAMETER VALUE	
4891	781F	CD	2E 78	CALL MPYTSZ ;MULITPLY BY SIZE	
4892	7822	EB	.	XCHG ;DE = PARAMETER	
4893	7823	E1	.	POP H ;RECALL PARAMETER POINTER	
4894	7824	73	.	MOV M,E ;STORE UPDATED LSBYTE	
4895	7825	23	.	INX H	
4896	7826	72	.	MOV M,D ;STORE UPDATED MSBYTE	
4897	7827	2B	.	DCX H ;UPDATE POINTER	
4898	7828	2B	.	DCX H	
4899	7829	2B	.	DCX H	
4900	782A	F1	.	POP PSW ;RECALL COUNT	
4901	782B	C3	17 78	JMP MPYALL ;DO NEW PARAMETER	
4902	782E	.	.	;*****	
4903	782E	.	.	; MPYTSZ--MULITPLY HL BY TEXT SIZE	
4904	782E	.	.	; EXIT HL = HL * TEXT SIZE	
4905	782E	.	.	; DE, A DESTROYED	
4906	782E	.	.	;*****	
4907	782E	.	.	MPYTSZ EQU \$	
4908	782E	3A	DA FB	LDA TXMAG ;LOAD SIZE	
4909	7831	.	.	MPY1 EQU \$	
4910	7831	EB	.	XCHG ;DE = PARAMETER	
4911	7832	21	00 00	LXI H,0	
4912	7835	.	.	MPT010 EQU \$	
4913	7835	19	.	DAD D ;ADD PARAMETER	
4914	7836	3D	.	DCR A ;UPDATE COUNT	
4915	7837	F2	35 78	JP MPT010 ;LOOP UNTIL DONE	
4916	783A	C9	.	RET	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 136
4918	783B	. . .	;*****	
4919	783B	. . .	; PUTCHR--PUT CHAR INTO GRAPHICS AT CURRENT	
4920	783B	. . .	; POINT. CHARACTER WILL NOT BE DRAWN IF NOT	
4921	783B	. . .	; COMPLETELY ON SCREEN. CURRENT POINT IS	
4922	783B	. . .	; UPDATED FOR NEXT CHAR EVEN IF CHAR ISNT	
4923	783B	. . .	; DRAWN.	
4924	783B	. . .	; ENTRY ZDCHAR = CHARACTER TO BE DISPLAYED	
4925	783B	. . .	;*****	
4926	783B	. . .	PUTCHR EQU \$	
4927	783B	CD F4 78	CALL PCH050 ;DO CURSOR UPDATE	
4928	783E	3A 97 90	LDA GFLGS6 ;IN CENTER OR RT. JUSTIFY MO	
4929	7841	E6 18 .	ANI RTJUST+CNTR	
4930	7843	C2 DD 78	JNZ PCH020 ;YES, DONT DISPLAY CHAR	
4931	7846	. . .	PCH007 EQU \$	
4932	7846	CD 02 98	CALL CHROK ;SEE IF CHAR WILL FIT ON SCR	
4933	7849	FA 6F 78	JM PCH010 ;NO, JUST UPDATE POSITION	
4934	784C	. . .	; COMPUTE POINTER TO CHARACTER PATTERNS IN TABLE	
4935	784C	3A 89 FF	LDA ZDCHAR ;FETCH THE CHARACTER	
4936	784F	D6 20 .	SUI 40Q ;COMPENSATE FOR CONTROL CODE	
4937	7851	F8 . .	RM ;DONT PROCESS IF CONTROL COD	
4938	7852	. . .	; DONT PRINT DEL CHAR (177B)	
4939	7852	FE 5F .	CPI 137Q ;IS IT RUBOUT?	
4940	7854	D0 . .	RNC ;YES, DONT PRINT	
4941	7855	3C . .	INR A	
4942	7856	6F . .	MOV L,A	
4943	7857	26 00 .	MVI H,0 ;HL = CHAR	
4944	7859	. . .	; USE CHAR AS INDEX TO TABLE. MULTIPLY BY 10	
4945	7859	. . .	; (10 BYTES/CHAR)	
4946	7859	29 . .	DAD H ;CHAR * 2	
4947	785A	5D . .	MOV E,L ;SAVE 2 * CHAR	
4948	785B	54 . .	MOV D,H	
4949	785C	29 . .	DAD H ;4 * CHAR	
4950	785D	29 . .	DAD H ;8 * CHAR	
4951	785E	19 . .	DAD D ;10 * CHAR	
4952	785F	. . .	; HL = POINTER TO FIRST BYTE OF NEXT CHAR	
4953	785F	. . .	; WANT LAST BYTE OF CURRENT CHAR	
4954	785F	2B . .	DCX H	
4955	7860	11 18 79	LXI D,CHRTAB ;BASE OF CHARACTERS	
4956	7863	19 . .	DAD D	
4957	7864	22 81 90	SHLD CHPAT ;SAVE POINTER	
4958	7867	. . .	; DRAW THE CHARACTER	
4959	7867	CD 3A 98	CALL CHFILL	
4960	786A	21 67 FF	LXI H,ZCAFLG ;SET THE CURSOR ADVANCE FLAG	
4961	786D	36 01 .	MVI M,1	
4962	786F	. . .	; UPDATE THE CURRENT POINT	
4963	786F	. . .	PCH010 EQU \$	
4964	786F	2A 89 90	LHLD XCHINC ;UPDATE X COORD	
4965	7872	EB . .	XCHG	
4966	7873	2A DE 90	LHLD XCURR	
4967	7876	19 . .	DAD D	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 137
4968	7877	22	DA	90	SHLD XNEW	
4969	787A	2A	87	90	LHLD YCHINC	;UPDATE Y COORD
4970	787D	EB	.	.	XCHG	
4971	787E	2A	DC	90	LHLD YCURR	
4972	7881	19	.	.	DAD D	
4973	7882	22	D8	90	SHLD YNEW	
4974	7885	.	.	.	PCH1 EQU \$	
4975	7885	.	.	.	; DONT UPDATE START OF LINE	
4976	7885	3E	04	.	MVI A,NOSOL	
4977	7887	CD	5A	A2	CALL STFLG6	
4978	788A	.	.	.	PCH2 EQU \$; (UPDATE SOL)
4979	788A	CD	32	98	CALL CPUPDA	;UPDATE THE CURRENT POINT
4980	788D	.	.	.	; UPDATE THE CURSOR POSITION	
4981	788D	.	.	.	; CURSOR TRACKS CURRENT POINT IF IT IS ON	
4982	788D	3A	80	90	LDA GFLGS3	;IS CURSOR ON?
4983	7890	E6	80	.	ANI WANTGC	
4984	7892	CA	9D	78	JZ PCH015	;NO, DONE
4985	7895	3A	97	90	LDA GFLGS6	;IN LABEL MODE?
4986	7898	E6	80	.	ANI LABEL	
4987	789A	CC	E6	A3	CZ MOVEGC	;NO, MOVE CURSOR TO PEN
4988	789D	.	.	.	PCH015 EQU \$	
4989	789D	.	.	.	; IF IN SCALED TEK MODE, DO AUTOLF IF PAST 512	
4990	789D	CD	5C	69	CALL CKSCLD	;IN SCALED TEK MODE?
4991	78A0	C8	.	.	RZ	;NO, DONE
4992	78A1	.	.	.	; IF AUTO PLOT LABEL, DONT DO AUTOLF	
4993	78A1	3A	96	90	LDA GFLGS7	;AUTO PLOT LABEL IN PROGRESS?
4994	78A4	E6	02	.	ANI APLABL	
4995	78A6	C0	.	.	RNZ	;YES, SKIP BOUNDS CHECK
4996	78A7	2A	DE	90	LHLD XCURR	;PAST 511 = RIGHT MARGIN
4997	78AA	11	FF	01	LXI D,511	;MAX SCALED X COORD
4998	78AD	CD	5F	89	CALL CHKMAX	
4999	78B0	D2	89	78	JNC PCH017	;NO, CHECK OTHER BOUNDRY
5000	78B3	CD	AF	9C	CALL XCR	;YES, DO CR/LF
5001	78B6	CD	F9	9C	CALL XLF	
5002	78B9	.	.	.	PCH017 EQU \$	
5003	78B9	.	.	.	; IF X COORD IS .LT. 0, SET IT TO 0	
5004	78B9	.	.	.	; (SCALED TEK ONLY)	
5005	78B9	0E	00	.	MVI C,0	;OUT OF BOUNDS FLAG
5006	78BB	2A	DE	90	LHLD XCURR	
5007	78BE	7C	.	.	MOV A,H	;IS X NEG?
5008	78BF	87	.	.	ORA A	
5009	78C0	F2	C7	78	JP PCH018	;NO
5010	78C3	21	00	00	LXI H,0	;YES, SET TO 0
5011	78C6	0C	.	.	INR C	;SET OUT OF BOUNDS FLAG
5012	78C7	.	.	.	PCH018 EQU \$	
5013	78C7	22	DA	90	SHLD XNEW	
5014	78CA	.	.	.	; TEST FOR Y GREATER THAN TOP OF SCREEN	
5015	78CA	2A	DC	90	LHLD YCURR	
5016	78CD	CD	69	A3	CALL YCHECK	;IN RANGE?
5017	78D0	F2	D4	78	JP PCH019	;YES

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 138
=====
5018      78D3      0C      .      .      INR      C      ;SET OUT OF BOUNDS FLAG
5019      78D4      .      .      .      PCH019 EQU $
5020      78D4      22      D8      90      SHLD      YNEW
5021      78D7      .      .      .      ; IF EITHER WAS OUT OF BOUNDS, UPDATE PEN POSITION
5022      78D7      79      .      .      MOV      A,C      ;TEST OUT OF BOUNDS FLAG
5023      78D8      B7      .      .      ORA      A
5024      78D9      C8      .      .      RZ      ;BOTH OK, DONE
5025      78DA      C3      32      98      JMP      CPUPDA      ;UPDATE PEN POSITION
5026      78DD      .      .      .      PCH020 EQU $
5027      78DD      .      .      .      ;IN RIGHT JUSTIFY OR CENTER MODE
5028      78DD      .      .      .      ;JUST STICK CHAR INTO BUFFER FOR NOW
5029      78DD      .      .      .      ; BUFFER IS DUMPED WHEN CR OR LF IS RECEIVED
5030      78DD      21      74      90      LXI      H,LBLCTR ;BUFFER FULL?
5031      78E0      7E      .      .      MOV      A,M
5032      78E1      FE      84      .      CPI      MAXLBL
5033      78E3      D0      .      .      RNC      ;YES, IGNORE CHAR
5034      78E4      5F      .      .      MOV      E,A
5035      78E5      16      00      .      MVI      D,0      ;DE = INDEX TO EMPTY SLOT
5036      78E7      3A      89      FF      LDA      ZDCHAR      ;FETCH THE CHAR
5037      78EA      FE      20      .      CPI      400      ;CONTROL CODE?
5038      78EC      D8      .      .      RC      ;YES, IGNORE
5039      78ED      34      .      .      INR      M      ;UPDATE COUNT
5040      78EE      21      0D      FB      LXI      H,LBLBUF ;BASE OF BUFFER
5041      78F1      19      .      .      DAD      D      ;HL = POINTER TO BUFFER SLOT
5042      78F2      77      .      .      MOV      M,A      ;STORE THE CHAR
5043      78F3      C9      .      .      RET
5044      78F4      .      .      .      PCH050 EQU $
5045      78F4      .      .      .      ; IS CURSOR ON?
5046      78F4      3A      80      90      LDA      GFLGS3
5047      78F7      E6      80      .      ANI      WANTGC
5048      78F9      C8      .      .      RZ      ;NO, DONT DO UPDATE
5049      78FA      .      .      .      ; IN LABEL MODE?
5050      78FA      3A      97      90      LDA      GFLGS6
5051      78FD      E6      80      .      ANI      LABEL
5052      78FF      C0      .      .      RNZ      ;YES, DONT DO UPDATE
5053      7900      .      .      .      ; IT IS ON, HAS IT MOVED?
5054      7900      3A      AE      90      LDA      GFLGS5
5055      7903      E6      40      .      ANI      GCM4
5056      7905      C8      .      .      RZ      ;NO, DONT DO UPDATE
5057      7906      .      .      .      ; CURSOR IS ON AND HAS MOVED, SET SOL AND CURRENT
5058      7906      .      .      .      ; PEN POSITION TO CURSOR POSITION
5059      7906      CD      46      A2      CALL      CLFLGS      ;CLEAR THE MOVED FLAG
5060      7909      2A      CF      90      LHLD      NEWGCX
5061      790C      22      DA      90      SHLD      XNEW      ;SET CURRENT POINT, SOL
5062      790F      2A      CD      90      LHLD      NEWGCY      ;TO CURSOR LOCATION BEFORE
5063      7912      22      D8      90      SHLD      YNEW      ;DRAWING CHAR OR CONTROL COD
5064      7915      C3      32      98      JMP      CPUPDA
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 139
=====
5066      7918      . . .      ;*****
5067      7918      . . .      ; CHARACTER PATTERNS--
5068      7918      . . .      ; STORED AS 7 BY 10 (10 BYTES)
5069      7918      . . .      ;*****
5070      7918      . . .      CHRTAB EQU $
5071      7918      00 00 00      DB 000Q,000Q,000Q,000Q,000Q ;SPACE
5072      791D      00 00 00      DB 000Q,000Q,000Q,000Q,000Q
5073      7922      00 10 10      DB 000Q,020Q,020Q,020Q,020Q ;!
5074      7927      10 00 10      DB 020Q,000Q,020Q,000Q,000Q
5075      792C      00 28 28      DB 000Q,050Q,050Q,050Q,000Q ;"
5076      7931      00 00 00      DB 000Q,000Q,000Q,000Q,000Q
5077      7936      00 00 28      DB 000Q,000Q,050Q,174Q,050Q ;#
5078      793B      7C 28 00      DB 174Q,050Q,000Q,000Q,000Q
5079      7940      00 10 3C      DB 000Q,020Q,074Q,120Q,070Q ;$
5080      7945      14 78 10      DB 024Q,170Q,020Q,000Q,000Q
5081      794A      00 60 64      DB 000Q,140Q,144Q,010Q,020Q ;%
5082      794F      20 0C 0C      DB 040Q,014Q,014Q,000Q,000Q
5083      7954      00 20 50      DB 000Q,040Q,120Q,120Q,040Q ;&
5084      7959      54 48 34      DB 124Q,110Q,064Q,000Q,000Q
5085      795E      00 10 10      DB 000Q,020Q,020Q,040Q,000Q ;'
5086      7963      00 00 00      DB 000Q,000Q,000Q,000Q,000Q
5087      7968      00 08 10      DB 000Q,010Q,020Q,020Q,020Q ;(
5088      796D      10 10 08      DB 020Q,020Q,010Q,000Q,000Q
5089      7972      00 20 10      DB 000Q,040Q,020Q,020Q,020Q ;)
5090      7977      10 10 20      DB 020Q,020Q,040Q,000Q,000Q
5091      797C      00 00 28      DB 000Q,000Q,050Q,020Q,174Q ;*
5092      7981      10 28 00      DB 020Q,050Q,000Q,000Q,000Q
5093      7986      00 00 10      DB 000Q,000Q,020Q,020Q,174Q ;+
5094      798B      10 10 00      DB 020Q,020Q,000Q,000Q,000Q
5095      7990      00 00 00      DB 000Q,000Q,000Q,000Q,000Q ;,
5096      7995      00 18 18      DB 000Q,030Q,030Q,010Q,020Q
5097      799A      00 00 00      DB 000Q,000Q,000Q,000Q,174Q ;-
5098      799F      00 00 00      DB 000Q,000Q,000Q,000Q,000Q
5099      79A4      00 00 00      DB 000Q,000Q,000Q,000Q,000Q ;.
5100      79A9      00 18 18      DB 000Q,030Q,030Q,000Q,000Q
5101      79AE      00 00 04      DB 000Q,000Q,004Q,010Q,020Q ;/
5102      79B3      20 40 00      DB 040Q,100Q,000Q,000Q,000Q
5103      79B8      00 38 44      DB 000Q,070Q,104Q,114Q,124Q ;0
5104      79BD      64 44 38      DB 144Q,104Q,070Q,000Q,000Q
5105      79C2      00 10 30      DB 000Q,020Q,060Q,020Q,020Q ;1
5106      79C7      10 10 38      DB 020Q,020Q,070Q,000Q,000Q
5107      79CC      00 38 44      DB 000Q,070Q,104Q,004Q,030Q ;2
5108      79D1      20 40 7C      DB 040Q,100Q,174Q,000Q,000Q
5109      79D6      00 38 44      DB 000Q,070Q,104Q,004Q,030Q ;3
5110      79DB      04 44 38      DB 004Q,104Q,070Q,000Q,000Q
5111      79E0      00 08 18      DB 000Q,010Q,030Q,050Q,110Q ;4
5112      79E5      7C 08 08      DB 174Q,010Q,010Q,000Q,000Q
5113      79EA      00 7C 40      DB 000Q,174Q,100Q,100Q,170Q ;5
5114      79EF      04 44 38      DB 004Q,104Q,070Q,000Q,000Q
5115      79F4      00 18 20      DB 000Q,030Q,040Q,100Q,170Q ;6
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
5116	79F9	44	44	38	DB 104Q,104Q,070Q,000Q,000Q	
5117	79FE	00	7C	04	DB 000Q,174Q,004Q,010Q,020Q	;7
5118	7A03	20	20	20	DB 040Q,040Q,040Q,000Q,000Q	
5119	7A08	00	38	44	DB 000Q,070Q,104Q,104Q,070Q	;8
5120	7A0D	44	44	38	DB 104Q,104Q,070Q,000Q,000Q	
5121	7A12	00	38	44	DB 000Q,070Q,104Q,104Q,074Q	;9
5122	7A17	04	08	30	DB 004Q,010Q,060Q,000Q,000Q	
5123	7A1C	00	00	00	DB 000Q,000Q,000Q,030Q,030Q	::
5124	7A21	00	18	18	DB 000Q,030Q,030Q,000Q,000Q	::
5125	7A26	00	00	00	DB 000Q,000Q,000Q,030Q,030Q	::;
5126	7A2B	00	18	18	DB 000Q,030Q,030Q,010Q,020Q	::<
5127	7A30	00	08	10	DB 000Q,010Q,020Q,040Q,100Q	::<
5128	7A35	20	10	08	DB 040Q,020Q,010Q,000Q,000Q	
5129	7A3A	00	00	00	DB 000Q,000Q,000Q,174Q,000Q	::=
5130	7A3F	7C	00	00	DB 174Q,000Q,000Q,000Q,000Q	::=
5131	7A44	00	40	20	DB 000Q,100Q,040Q,020Q,010Q	::>
5132	7A49	10	20	40	DB 020Q,040Q,100Q,000Q,000Q	::>
5133	7A4E	00	38	44	DB 000Q,070Q,104Q,104Q,010Q	::?
5134	7A53	10	00	10	DB 020Q,000Q,020Q,000Q,000Q	::?
5135	7A58	00	38	44	DB 000Q,070Q,104Q,134Q,124Q	::@
5136	7A5D	5C	40	38	DB 134Q,100Q,070Q,000Q,000Q	::@
5137	7A62	00	38	44	DB 000Q,070Q,104Q,104Q,104Q	::A
5138	7A67	7C	44	44	DB 174Q,104Q,104Q,000Q,000Q	::A
5139	7A6C	00	78	24	DB 000Q,170Q,044Q,044Q,070Q	::B
5140	7A71	24	24	78	DB 044Q,044Q,170Q,000Q,000Q	::B
5141	7A76	00	38	44	DB 000Q,070Q,104Q,100Q,100Q	::C
5142	7A7B	40	44	38	DB 100Q,104Q,070Q,000Q,000Q	::C
5143	7A80	00	78	24	DB 000Q,170Q,044Q,044Q,044Q	::D
5144	7A85	24	24	78	DB 044Q,044Q,170Q,000Q,000Q	::D
5145	7A8A	00	7C	40	DB 000Q,174Q,100Q,100Q,160Q	::E
5146	7A8F	40	40	7C	DB 100Q,100Q,174Q,000Q,000Q	::E
5147	7A94	00	7C	40	DB 000Q,174Q,100Q,100Q,160Q	::F
5148	7A99	40	40	40	DB 100Q,100Q,100Q,000Q,000Q	::F
5149	7A9E	00	38	44	DB 000Q,070Q,104Q,100Q,100Q	::G
5150	7AA3	5C	44	38	DB 134Q,104Q,070Q,000Q,000Q	::G
5151	7AA8	00	44	44	DB 000Q,104Q,104Q,104Q,174Q	::H
5152	7AAD	44	44	44	DB 104Q,104Q,104Q,000Q,000Q	::H
5153	7AB2	00	38	10	DB 000Q,070Q,020Q,020Q,020Q	::I
5154	7AB7	10	10	38	DB 020Q,020Q,070Q,000Q,000Q	::I
5155	7ABC	00	04	04	DB 000Q,004Q,004Q,004Q,004Q	::J
5156	7AC1	04	44	38	DB 004Q,104Q,070Q,000Q,000Q	::J
5157	7AC6	00	44	48	DB 000Q,104Q,110Q,120Q,140Q	::K
5158	7ACB	50	48	44	DB 120Q,110Q,104Q,000Q,000Q	::K
5159	7AD0	00	40	40	DB 000Q,100Q,100Q,100Q,100Q	::L
5160	7AD5	40	40	7C	DB 100Q,100Q,174Q,000Q,000Q	::L
5161	7ADA	00	44	6C	DB 000Q,104Q,154Q,124Q,124Q	::M
5162	7ADF	44	44	44	DB 104Q,104Q,104Q,000Q,000Q	::M
5163	7AE4	00	44	44	DB 000Q,104Q,104Q,144Q,124Q	::N
5164	7AE9	4C	44	44	DB 114Q,104Q,104Q,000Q,000Q	::N
5165	7AEE	00	38	44	DB 000Q,070Q,104Q,104Q,104Q	::O

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 141
5166	7AF3	44	44	38	DB 104Q,104Q,070Q,000Q,000Q	
5167	7AF8	00	78	44	DB 000Q,170Q,104Q,104Q,170Q	;P
5168	7AFD	40	40	40	DB 100Q,100Q,100Q,000Q,000Q	
5169	7B02	00	38	44	DB 000Q,070Q,104Q,104Q,104Q	;Q
5170	7B07	54	48	34	DB 124Q,110Q,064Q,000Q,000Q	
5171	7B0C	00	78	44	DB 000Q,170Q,104Q,104Q,170Q	;R
5172	7B11	50	48	44	DB 120Q,110Q,104Q,000Q,000Q	
5173	7B16	00	38	44	DB 000Q,070Q,104Q,040Q,020Q	;S
5174	7B1B	08	44	38	DB 010Q,104Q,070Q,000Q,000Q	
5175	7B20	00	7C	10	DB 000Q,174Q,020Q,020Q,020Q	;T
5176	7B25	10	10	10	DB 020Q,020Q,020Q,000Q,000Q	
5177	7B2A	00	44	44	DB 000Q,104Q,104Q,104Q,104Q	;U
5178	7B2F	44	44	38	DB 104Q,104Q,070Q,000Q,000Q	
5179	7B34	00	44	44	DB 000Q,104Q,104Q,104Q,050Q	;V
5180	7B39	28	10	10	DB 050Q,020Q,020Q,000Q,000Q	
5181	7B3E	00	44	44	DB 000Q,104Q,104Q,104Q,124Q	;W
5182	7B43	54	54	28	DB 124Q,124Q,050Q,000Q,000Q	
5183	7B48	00	44	44	DB 000Q,104Q,104Q,050Q,020Q	;X
5184	7B4D	28	44	44	DB 050Q,104Q,104Q,000Q,000Q	
5185	7B52	00	44	44	DB 000Q,104Q,104Q,050Q,020Q	;Y
5186	7B57	10	10	10	DB 020Q,020Q,020Q,000Q,000Q	
5187	7B5C	00	7C	04	DB 000Q,174Q,004Q,010Q,020Q	;Z
5188	7B61	20	40	7C	DB 040Q,100Q,174Q,000Q,000Q	
5189	7B66	00	38	20	DB 000Q,070Q,040Q,040Q,040Q	;[
5190	7B6B	20	20	38	DB 040Q,040Q,070Q,000Q,000Q	
5191	7B70	00	00	40	DB 000Q,000Q,100Q,040Q,020Q	;\ &
5192	7B75	08	04	00	DB 010Q,004Q,000Q,000Q,000Q	
5193	7B7A	00	38	08	DB 000Q,070Q,010Q,010Q,010Q	;] &
5194	7B7F	08	08	38	DB 010Q,010Q,070Q,000Q,000Q	
5195	7B84	00	10	28	DB 000Q,020Q,050Q,104Q,000Q	;^ &
5196	7B89	00	00	00	DB 000Q,000Q,000Q,000Q,000Q	
5197	7B8E	00	00	00	DB 000Q,000Q,000Q,000Q,000Q	;_ &
5198	7B93	00	00	00	DB 000Q,000Q,000Q,000Q,174Q	
5199	7B98	00	20	10	DB 000Q,040Q,020Q,010Q,000Q	;@ &
5200	7B9D	00	00	00	DB 000Q,000Q,000Q,000Q,000Q	
5201	7BA2	00	00	00	DB 000Q,000Q,000Q,070Q,004Q	;A &
5202	7BA7	3C	44	3E	DB 074Q,104Q,076Q,000Q,000Q	
5203	7BAC	00	40	40	DB 000Q,100Q,100Q,170Q,104Q	;B &
5204	7BB1	44	44	78	DB 104Q,104Q,170Q,000Q,000Q	
5205	7BB6	00	00	00	DB 000Q,000Q,000Q,070Q,100Q	;C &
5206	7BBB	40	40	38	DB 100Q,100Q,070Q,000Q,000Q	
5207	7BC0	00	04	04	DB 000Q,004Q,004Q,074Q,104Q	;D &
5208	7BC5	44	44	3C	DB 104Q,104Q,074Q,000Q,000Q	
5209	7BCA	00	00	00	DB 000Q,000Q,000Q,070Q,104Q	;E &
5210	7BCF	7C	40	38	DB 174Q,100Q,070Q,000Q,000Q	
5211	7BD4	00	0C	10	DB 000Q,014Q,020Q,074Q,020Q	;F &
5212	7BD9	10	10	10	DB 020Q,020Q,020Q,000Q,000Q	
5213	7BDE	00	00	00	DB 000Q,000Q,000Q,070Q,104Q	;G &
5214	7BE3	44	44	3C	DB 104Q,104Q,074Q,004Q,070Q	
5215	7BE8	00	40	40	DB 000Q,100Q,100Q,170Q,104Q	;H &

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 142
5216	7BED	44	44	44	DB 104Q,104Q,104Q,000Q,000Q	
5217	7BF2	00	10	00	DB 000Q,020Q,000Q,060Q,020Q	;I
5218	7BF7	10	10	38	DB 020Q,020Q,070Q,000Q,000Q	
5219	7BFC	00	08	00	DB 000Q,010Q,000Q,010Q,010Q	;J
5220	7C01	08	08	08	DB 010Q,010Q,010Q,110Q,060Q	
5221	7C06	00	40	40	DB 000Q,100Q,100Q,110Q,120Q	;K
5222	7C08	60	50	48	DB 140Q,120Q,110Q,000Q,000Q	
5223	7C10	00	30	10	DB 000Q,060Q,020Q,020Q,020Q	;L
5224	7C15	10	10	38	DB 020Q,020Q,070Q,000Q,000Q	
5225	7C1A	00	00	00	DB 000Q,000Q,000Q,170Q,124Q	;M
5226	7C1F	54	54	54	DB 124Q,124Q,124Q,000Q,000Q	
5227	7C24	00	00	00	DB 000Q,000Q,000Q,130Q,144Q	;N
5228	7C29	44	44	44	DB 104Q,104Q,104Q,000Q,000Q	
5229	7C2E	00	00	00	DB 000Q,000Q,000Q,070Q,104Q	;O
5230	7C33	44	44	38	DB 104Q,104Q,070Q,000Q,000Q	
5231	7C38	00	00	00	DB 000Q,000Q,000Q,170Q,104Q	;P
5232	7C3D	44	44	78	DB 104Q,104Q,170Q,100Q,100Q	
5233	7C42	00	00	00	DB 000Q,000Q,000Q,074Q,104Q	;Q
5234	7C47	44	44	3C	DB 104Q,104Q,074Q,004Q,004Q	
5235	7C4C	00	00	00	DB 000Q,000Q,000Q,054Q,060Q	;R
5236	7C51	20	20	20	DB 040Q,040Q,040Q,000Q,000Q	
5237	7C56	00	00	00	DB 000Q,000Q,000Q,074Q,100Q	;S
5238	7C58	38	04	78	DB 070Q,004Q,170Q,000Q,000Q	
5239	7C60	00	10	10	DB 000Q,020Q,020Q,074Q,020Q	;T
5240	7C65	10	10	18	DB 020Q,020Q,030Q,000Q,000Q	
5241	7C6A	00	00	00	DB 000Q,000Q,000Q,104Q,104Q	;U
5242	7C6F	44	44	3C	DB 104Q,104Q,074Q,000Q,000Q	
5243	7C74	00	00	00	DB 000Q,000Q,000Q,104Q,104Q	;V
5244	7C79	28	28	10	DB 050Q,050Q,020Q,000Q,000Q	
5245	7C7E	00	00	00	DB 000Q,000Q,000Q,104Q,104Q	;W
5246	7C83	54	54	28	DB 124Q,124Q,050Q,000Q,000Q	
5247	7C88	00	00	00	DB 000Q,000Q,000Q,104Q,050Q	;X
5248	7C8D	10	28	44	DB 020Q,050Q,104Q,000Q,000Q	
5249	7C92	00	00	00	DB 000Q,000Q,000Q,104Q,104Q	;Y
5250	7C97	44	28	10	DB 104Q,050Q,020Q,040Q,100Q	
5251	7C9C	00	00	00	DB 000Q,000Q,000Q,174Q,010Q	;Z
5252	7CA1	10	20	7C	DB 020Q,040Q,174Q,000Q,000Q	
5253	7CA6	00	0C	10	DB 000Q,014Q,020Q,020Q,140Q	;[
5254	7CAB	10	10	0C	DB 020Q,020Q,014Q,000Q,000Q	
5255	7CB0	00	10	10	DB 000Q,020Q,020Q,020Q,000Q	;\ ,
5256	7CB5	10	10	10	DB 020Q,020Q,020Q,000Q,000Q	
5257	7CBA	00	60	10	DB 000Q,140Q,020Q,020Q,014Q	;] ,
5258	7CBF	10	10	60	DB 020Q,020Q,140Q,000Q,000Q	
5259	7CC4	00	20	54	DB 000Q,040Q,124Q,010Q,000Q	;, ,
5260	7CC9	00	00	00	DB 000Q,000Q,000Q,000Q,000Q	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 143
5262	7CCE	.	.	*****	
5263	7CCE	.	.	; MAIN CODE OVERFLOW OCCUPIES 76400B TO 7777B	
5264	7CCE	.	.	*****	
5265	7CCE	.	.	*****	
5266	7CCE	.	.	ORG 114000Q ;SET ORIGIN TO 38K	
5267	9800	.	.	ZBRK4 EQU \$	
5268	9800	54	.	DB VERSN ;SET ROM PRESENT FLAGS	
5269	9801	98	.	DB ZBRK4/256	
5270	9802	.	.	*****	
5271	9802	.	.	*****	
5272	9802	.	.	; CHROK--SEE IF CHARACTER WILL BE COMPLETELY	
5273	9802	.	.	; ON SCREEN.	
5274	9802	.	.	; EXIT M FLAG => NOT ON SCREEN	
5275	9802	.	.	*****	
5276	9802	.	.	CHROK EQU \$	
5277	9802	.	.	; FIRST CHECK X CCORDS	
5278	9802	2A	DE 90	LHLD XCURR ;CURRENT POINT = LL OF CHAR	
5279	9805	EB	.	XCHG ;DE = LL CORNER OF CHAR	
5280	9806	2A	77 90	LHLD XCHADJ ;ADJUST FOR TOP OR MIDDLE	
5281	9809	19	.	DAD D ;OF CHAR	
5282	980A	EB	.	XCHG	
5283	980B	2A	8D 90	LHLD XCHSIZ	
5284	980E	19	.	DAD D ;HL = UR CORNER OF CHAR	
5285	980F	CD	60 A3	CALL XCHECK ;SEE OF UR IN BOUNDS	
5286	9812	F8	.	RM ;DONE IF NOT	
5287	9813	EB	.	XCHG ;HL = LL CORNER	
5288	9814	CD	60 A3	CALL XCHECK ;SEE IF LL IN BOUNDS	
5289	9817	F8	.	RM ;DONE IF NOT	
5290	9818	.	.	; CHECK Y COORDINATES	
5291	9818	2A	DC 90	LHLD YCURR ;LL CORNER OF CHAR	
5292	981B	EB	.	XCHG ;DE = LL OF CHAR	
5293	981C	2A	75 90	LHLD YCHADJ ;ADJUST FOR TOP OR MIDDLE	
5294	981F	19	.	DAD D ;OF CHAR	
5295	9820	EB	.	XCHG	
5296	9821	2A	8B 90	LHLD YCHSIZ	
5297	9824	19	.	DAD D ;HL = UR	
5298	9825	CD	69 A3	CALL YCHECK ;CHECK UPPER RIGHT	
5299	9828	F8	.	RM ;DONE IF NOT IN BOUNDS	
5300	9829	EB	.	XCHG	
5301	982A	C3	69 A3	JMP YCHECK ;TEST LL	
5302	982D	.	.	*****	
5303	982D	.	.	; CPUPDA--UPDATE THE CURRENT POINT BY DOING A	
5304	982D	.	.	; MOVE TO NEW POINT	
5305	982D	.	.	*****	
5306	982D	.	.	CPUPD1 EQU \$	
5307	982D	.	.	; DONT CHANGE SOL POINT	
5308	982D	3E	04 .	MVI A,NOSOL	
5309	982F	CD	5A A2	CALL STFLG6	
5310	9832	.	.	CPUPDA EQU \$	
5311	9832	3E	01 .	MVI A,MOVE ;DO A MOVE TO NEWPOINT	

13255
2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

```
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 144
5312	9834	CD 26 A2	CALL STFLG1	
5313	9837	C3 E9 65	JMP VECTRO	

```
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 145
5315	983A	.	.	*****	
5316	983A	.	.	; CHFILL--DRAW CHARACTER AT CURRENT POINT	
5317	983A	.	.	; ALL PARAMETERS HAVE BEEN PREVIOUSLY SET	
5318	983A	.	.	*****	
5319	983A	.	.	CHFILL EQU \$	
5320	983A	.	.	; SUPPRESS CURSOR NOW	
5321	983A	.	.	; BOTH CHFILL AND CURSOR USE TEMP VARIABLES	
5322	983A	3E	08	MVI A,TIMSUP	
5323	983C	CD	8A	9E	CALL SUPRGC
5324	983F	21	6A	90	LXI H,CNT1 ;COUNTER FOR 10 PATTERN
5325	9842	36	0A	.	MVI M,10 ;BYTES
5326	9844	.	.	CHFIL1 EQU \$	
5327	9844	3A	DA	FB	LDA TXMAG ;COUNTER FOR # OF TIMES PAT
5328	9847	32	69	90	STA CNT2 ;IS REPEATED = TEXT SIZE
5329	984A	2A	DC	90	LHLD YCURR ;GET INITIAL WA
5330	984D	EB	.	.	XCHG ;ADJUST FOR TOP OR
5331	984E	2A	75	90	LHLD YCHADJ ;MIDDLE OF CHAR
5332	9851	19	.	.	DAD D
5333	9852	CD	5B	67	CALL MPY45
5334	9855	E5	.	.	PUSH H
5335	9856	2A	DE	90	LHLD XCURR ;SAVE INITIAL ADDR ON STACK
5336	9859	EB	.	.	XCHG ;ADJUST FOR TOP OR
5337	985A	2A	77	90	LHLD XCHADJ ;MIDDLE OF CHAR
5338	985D	19	.	.	DAD D
5339	985E	E5	.	.	PUSH H
5340	985F	.	.	.	; CONTROLLER IS IDLE FROM CURSOR SUPPRESS
5341	985F	.	.	.	; LOAD CONTROLLER WITH CONSTANT PARAMETERS
5342	985F	CD	0F	A4	CALL VSETUP ;SET UP CONSTANT PARAMETERS
5343	9862	2A	D8	FB	LHLD CHLEN ;LOAD VECTOR LENGTH
5344	9865	22	12	89	SHLD DC ;SEND DOT COUNT
5345	9868	2A	93	90	LHLD CFM1 ;SEND M1
5346	986B	22	1A	89	SHLD M1
5347	986E	3A	95	90	LDA CFM1+2
5348	9871	32	18	89	STA SIGNM1
5349	9874	3A	DA	FB	LDA TXMAG ;SET PRESCALER
5350	9877	32	21	89	STA SCALER
5351	987A	3A	B5	90	LDA CURMOD ;TURN SAMPLE AND PATTERN
5352	987D	F6	24	.	ORI 440 ;PATTERN ON
5353	987F	32	41	89	STA HCEJK
5354	9882	D1	.	.	POP D ;DE = X COORD
5355	9883	E1	.	.	POP H ;HL = Y COORD
5356	9884	.	.	CHF010 EQU \$	
5357	9884	E5	.	.	PUSH H ;SAVE Y COORD
5358	9885	D5	.	.	PUSH D ;SAVE X
5359	9886	CD	6F	67	CALL GETWA ;GET WRITE ADDRESS
5360	9889	22	0E	89	SHLD LSBWA ;SEND WA 0-11
5361	988C	32	0C	89	STA MSBWA ;SEND WA 12-17
5362	988F	2A	81	90	LHLD CHPAT ;LOAD POINTER TO CHAR PATTERN
5363	9892	7E	.	.	MOV A,M ;FETCH CURRENT PATTERN
5364	9893	32	40	89	STA PATTERN ;SEND TO HW

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 146
5365	9896	CD	22	67	CALL HWGO ;START HW GOING
5366	9899	.	.	.	; SEE IF TIME FOR NEW PATTERN
5367	9899	21	69	90	LXI H,CNT2 ;CONTINUE REPEATING THIS
5368	989C	35	.	.	DCR M ;PATTERN?
5369	989D	F2	84	98	JP CHF020 ;YES
5370	98A0	.	.	.	; HAVE ALL PATTERN BYTES BEEN USED?
5371	98A0	21	6A	90	LXI H,CNT1 ;ALL 10 PAT BYTES USED?
5372	98A3	35	.	.	DCR M
5373	98A4	CA	C5	98	JZ CHF030 ;YES--DONE
5374	98A7	2A	81	90	LHLD CHPAT ;NO, UPDATE PATTERN POINTER
5375	98AA	2B	.	.	DCX H
5376	98AB	22	81	90	SHLD CHPAT
5377	98AE	3A	DA	FB	LDA TXMAG ;UPDATE PATTERN REPEAT
5378	98B1	32	69	90	STA CNT2 ;COUNTER
5379	98B4	.	.	.	CHF020 EQU \$
5380	98B4	.	.	.	; UPDATE THE WRITE ADDRESS FOR THE NEXT VECTOR
5381	98B4	D1	.	.	POP D ;GET X
5382	98B5	2A	91	90	LHLD CFXINC ;X CHAR FILL INCREMENT
5383	98B8	19	.	.	DAD D
5384	98B9	EB	.	.	XCHG ;DE = NEW X
5385	98BA	C1	.	.	POP B ;BC = Y
5386	98BB	2A	8F	90	LHLD CFYINC ;Y CHAR FILL INC
5387	98BE	09	.	.	DAD B ;HL = NEW Y
5388	98BF	.	.	.	; HL = Y COORD, DE = X COORD FOR NEXT VECTOR
5389	98BF	CD	87	A2	CALL WAIT ;WAIT FOR HW TO FINISH
5390	98C2	C3	84	98	JMP CHF010 ;DO NEXT VECTOR
5391	98C5	.	.	.	CHF030 EQU \$
5392	98C5	E1	.	.	POP H ;CLEAN UP STACK
5393	98C6	E1	.	.	POP H
5394	98C7	3E	08	.	MVI A,NEWWA ;SET FLAGS FOR NEXT VECTOR
5395	98C9	CD	26	A2	CALL STFLG1 ;MUST USE NEW WA
5396	98CC	3E	08	.	MVI A,DWFRST ;MUST DRAW FIRST DOT
5397	98CE	C3	40	A2	JMP STFLG5

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 147
=====
5399     98D1      . . .      ;*****
5400     98D1      . . .      ; SNDBUF--SEND BUFFER OF CHARACTERS TO GRAPHICS
5401     98D1      . . .      ; ENTRY HL = POINTER TO FIRST CHAR
5402     98D1      . . .      ; A = NUMBER OF CHARS
5403     98D1      . . .      ;*****
5404     98D1      . . .      SNDBUF EQU $
5405     98D1      . . .      ; ADJUST CURRENT POINT FOR FIRST CHAR SO THINGS
5406     98D1      . . .      ; WILL LINE UP
5407     98D1      . . .      ; ADJUST CURRENT POINT BY BACKSPACING BY NO. OF
5408     98D1      . . .      ; CHARS (FOR RT. JUST) OR 1/2 NO. OF CHARS (FOR
5409     98D1      . . .      ; CENTERING)
5410     98D1      E5 . . .      PUSH H          ;SAVE BUFFER POINTER
5411     98D2      F5 . . .      PUSH PSW       ;SAVE NO. OF CHARS
5412     98D3      4F . . .      MOV C,A        ;SAVE COUNT
5413     98D4      3A 97 90      LDA GFLGS6     ;IN CENTER OR RIGHT JUST MOD
5414     98D7      E6 18 .      ANI CNTR+RTJUST
5415     98D9      CA FC 98      JZ SBF025     ;NO, LEAVE CURR POINT AS IS
5416     98DC      79 . . .      MOV A,C        ;RECALL COUNT
5417     98DD      2A 89 90      LHL D XCHINC   ;GET X SPACING ADJUSTMENT
5418     98E0      CD 0F 99      CALL SBF050
5419     98E3      . . .      ; DE = X BACKSPACING INC
5420     98E3      2A DE 90      LHL D XCURR
5421     98E6      19 . . .      DAD D
5422     98E7      22 DA 90      SHLD XNEW      ;NEW X COORD
5423     98EA      . . .      ; UPDATE Y COORD
5424     98EA      F1 . . .      POP PSW        ;RECALL COUNT
5425     98EB      F5 . . .      PUSH PSW       ;SAVE IT
5426     98EC      2A 87 90      LHL D YCHINC   ;GET Y SPACING ADJUSTMENT
5427     98EF      CD 0F 99      CALL SBF050
5428     98F2      . . .      ; DE = Y BACKSPACING INC
5429     98F2      2A DC 90      LHL D YCURR
5430     98F5      19 . . .      DAD D
5431     98F6      22 08 90      SHLD YNEW      ;NEW Y COORD
5432     98F9      CD 2D 98      CALL CPUPD1    ;UPDATE CURRENT POINT
5433     98FC      . . .      SBF025 EQU $
5434     98FC      F1 . . .      POP PSW        ;RECALL COUNT
5435     98FD      E1 . . .      POP H          ;BUFFER POINTER
5436     98FE      . . .      SBF030 EQU $
5437     98FE      3D . . .      DCR A          ;ALL SENT?
5438     98FF      F8 . . .      RM             ;YES, DONE
5439     9900      F5 . . .      PUSH PSW       ;SAVE COUNT
5440     9901      7E . . .      MOV A,M        ;FETCH CHAR
5441     9902      32 89 FF      STA ZDCHAR     ;LEAVE WHERE PUTCHR WANTS IT
5442     9905      23 . . .      INX H          ;UPDATE BUFFER POINTER
5443     9906      E5 . . .      PUSH H        ;SAVE IT
5444     9907      CD 46 78      CALL PCH007    ;DRAW THE CHAR
5445     990A      E1 . . .      POP H          ;RECALL POINTER
5446     990B      F1 . . .      POP PSW        ;RECALL COUNT
5447     990C      C3 FE 98      JMP SBF030     ;DO THE NEXT CHAR
5448     990F      . . .      ;
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 148
5449	990F	. . .	;	
5450	990F	. . .	SBF050 EQU \$	
5451	990F	. . .	; COMPUTE BACKSPACING INCREMENT	
5452	990F	. . .	; HL = CHARACTER INCREMENT, A = CHAR COUNT	
5453	990F	3D . .	DCR A ;ADJUST COUNT FOR MPY1	
5454	9910	CD 31 78	CALL MPY1 ;MULTIPLY SPACING BY COUNT	
5455	9913	CD 09 A3	CALL NEGATE ;WANT TO BACKSPACE	
5456	9916	. . .	; DIVIDE BY 2 IF IN CENTERING MODE	
5457	9916	3A 97 90	LDA GFLGS6	
5458	9919	E6 08 .	ANI CNTR ;IN CENTERING MODE?	
5459	991B	C4 1A A3	CNZ DIVHL1 ;DIVIDE BY 2 IF SO	
5460	991E	EB . .	XCHG ;DE = BACKSPACE INCREMENT	
5461	991F	C9 . .	RET	
5462	9920	. . .	*****	
5463	9920	. . .	; SNDLBL--PUT CONTENTS OF LABEL BUFFER INTO	
5464	9920	. . .	; GRAPHICS USING CURRENT RT, LEFT OR CENTER	
5465	9920	. . .	; ENTRY HL = POINTER TO BUFFER	
5466	9920	. . .	; A = NUMBER OF CHARS	
5467	9920	. . .	*****	
5468	9920	. . .	SNDLB0 EQU \$;ENTRY FOR XCR, XLF	
5469	9920	3A 74 90	LDA LBLCTR ;ANYTHING IN BUFFER?	
5470	9923	B7 . .	ORA A	
5471	9924	C8 . .	RZ ;NO	
5472	9925	21 0D FB	LXI H,LBLBUF ;YES, SEND IT	
5473	9928	. . .	SNDLBL EQU \$	
5474	9928	EB . .	XCHG ;SAVE POINTER	
5475	9929	2A DE 90	LHLD XCURR ;SAVE THE CURRENT POINT	
5476	992C	E5 . .	PUSH H	
5477	992D	2A DC 90	LHLD YCURR	
5478	9930	E5 . .	PUSH H	
5479	9931	EB . .	XCHG ;RESTORE POINTER	
5480	9932	CD D1 98	CALL SNDBUF ;PRINT THE BUFFER	
5481	9935	AF . .	XRA A	
5482	9936	32 74 90	STA LBLCTR ;RESET CHAR COUNT	
5483	9939	E1 . .	POP H ;RESTORE CURRENT POINT	
5484	993A	22 D8 90	SHLD YNEW	
5485	993D	E1 . .	POP H	
5486	993E	22 DA 90	SHLD XNEW	
5487	9941	C3 32 98	JMP CPUPDA	

					PAGE 149	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS		
5489	9944	.	.	.	;*****	
5490	9944	.	.	.	; LORG--SET LABEL ORIGIN (1-9)	
5491	9944	.	.	.	;*****	
5492	9944	.	.	.	LOGR	EQU \$
5493	9944	CD	5C	69	CALL	CKSCLD ;DONT CHANGE IF SCALED TEK 0
5494	9947	C2	C1	99	JNZ	GEXIT
5495	994A	3E	09	.	MVI	A,9 ;MAX VALUE
5496	994C	CD	43	6D	CALL	GETPRM
5497	994F	C2	C1	99	JNZ	GEXIT ;IGNORE IF BAD
5498	9952	3D	.	.	DCR	A ;WANT 0-8, NOT 1-9
5499	9953	F4	59	99	CP	LOGR1 ;IGNORE IF IT WAS 0
5500	9956	C3	C1	99	JMP	GEXIT
5501	9959	.	.	.	LOGR1	EQU \$;(INTERNAL ENTRY)
5502	9959	32	D3	FB	STA	TXORG ;SAVE LOGR FOR STATUS
5503	995C	5F	.	.	MOV	E,A ;USE AS INDEX TO TABLE
5504	995D	16	00	.	MVI	D,0
5505	995F	21	76	99	LXI	H,LOGRTB ;BASE OF TABLE
5506	9962	19	.	.	DAD	D ;POINTER TO FLAGS
5507	9963	3A	97	90	LDA	GFLGS6 ;OLD LOGR FLAGS
5508	9966	E6	87	.	ANI	-1-RTJUST-CNTR-TOPCH-MIDCH ;DELETE OL
5509	9968	B6	.	.	ORA	M ;MERGE IN NEW
5510	9969	32	97	90	STA	GFLGS6 ;STORE NEW FLAGS
5511	996C	AF	.	.	XRA	A ;RESET LABEL COUNTER
5512	996D	32	74	90	STA	LBLCTR
5513	9970	3A	DB	FB	LDA	TANG ;COMPUTE NEW TEXT PARAMS
5514	9973	C3	48	76	JMP	ANGLE
5515	9976	.	.	.	;	
5516	9976	.	.	.	LOGRTB	EQU \$
5517	9976	00	.	.	DB	00
5518	9977	20	.	.	DB	400
5519	9978	40	.	.	DB	1000
5520	9979	08	.	.	DB	100
5521	997A	28	.	.	DB	500
5522	997B	48	.	.	DB	1100
5523	997C	10	.	.	DB	200
5524	997D	30	.	.	DB	600
5525	997E	50	.	.	DB	1200

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 150
5527	997F	.	.	*****	
5528	997F	.	.	; DEFAULT--SET GRAPHICS DEFAULTS	
5529	997F	.	.	*****	
5530	997F	.	.	DEFAULT EQU \$	
5531	997F	CD	85 99	CALL DEFLT1	
5532	9982	C3	C1 99	JMP GEXIT	
5533	9985	.	.	DEFLT1 EQU \$;INTERNAL ENTRY	
5534	9985	CD	18 6E	CALL ANVON1 ;TURN A/N VIDEO ON	
5535	9988	.	.	DEFLT2 EQU \$	
5536	9988	CD	82 88	CALL APLTOF ;TURN AUTO PLOT OFF	
5537	998B	CD	02 70	CALL TGCOF1 ;TURN THE CURSOR OFF	
5538	998E	CD	78 6E	CALL UNZOOM ;STOP ZOOMING	
5539	9991	.	.	; MUST SAVE TEK FLAGS AND STRAPS, PEN POSITION,	
5540	9991	.	.	; AND RESET FLAG	
5541	9991	3A	AD 90	LDA TKFLGS	
5542	9994	E6	43 .	ANI SCLD+UNSCLD+MARG1	
5543	9996	F5	.	PUSH PSW	
5544	9997	3A	C4 FB	LDA TEKPF ;SAVE TEK FULL STRAPS	
5545	999A	F5	.	PUSH PSW	
5546	999B	2A	DE 90	LHLD XCURR ;SAVE CURRENT POINT	
5547	999E	E5	.	PUSH H	
5548	999F	2A	DC 90	LHLD YCURR	
5549	99A2	E5	.	PUSH H	
5550	99A3	3A	96 90	LDA GFLGS7 ;SAVE RESET FLAG	
5551	99A6	E6	80 .	ANI RESET	
5552	99A8	F5	.	PUSH PSW	
5553	99A9	CD	D4 62	CALL HARD1 ;DO THE OTHER RESETS	
5554	99AC	F1	.	POP PSW ;RESTORE RESET FLAG	
5555	99AD	32	96 90	STA GFLGS7	
5556	99B0	E1	.	POP H ;RESTORE CURRENT POINT	
5557	99B1	22	DC 90	SHLD YCURR	
5558	99B4	E1	.	POP H	
5559	99B5	22	DE 90	SHLD XCURR	
5560	99B8	F1	.	POP PSW ;TEK STRAPS	
5561	99B9	32	C4 FB	STA TEKPF ;RESTORE TEK FULL STRAPS	
5562	99BC	F1	.	POP PSW ;TEK FLAGS	
5563	99BD	32	AD 90	STA TKFLGS	
5564	99C0	C9	.	RET	

				PAGE 151	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	
5566	99C1	.	.	.	;*****
5567	99C1	.	.	.	; GEXIT--TERMINATE GRAPHICS ESCAPE SEQUENCE IF
5568	99C1	.	.	.	; CAPITAL LETTER COMMAND RECEIVED
5569	99C1	.	.	.	;*****
5570	99C1	.	.	.	NOP EQU \$
5571	99C1	.	.	.	GEXIT EQU \$
5572	99C1	.	.	.	; FOR SYNCH, SET COMPUTE NEW WA FLAG
5573	99C1	3E	08	.	MVI A,NEWWA
5574	99C3	CD	26	A2	CALL STFLG1
5575	99C6	AF	.	.	XRA A ;CLEAR PARAMETER COUNT
5576	99C7	32	86	90	STA PRMDEX
5577	99CA	3E	39	.	MVI A,NIP+MINUS+HAVED+HAVEP
5578	99CC	CD	6D	A2	CALL CLFLG7 ;CLEAR PARAMETER FLAGS
5579	99CF	.	.	.	; SEE IF COMMAND WAS LOWER CASE
5580	99CF	3A	88	FF	LDA ZCHAR ;FETCH COMMAND
5581	99D2	E6	20	.	ANI 400 ;LOWER CASE?
5582	99D4	C0	.	.	RNZ ;YES, CONTINUE SEQUENCE
5583	99D5	CD	AB	9E	CALL ENAB0 ;RE-ENABLE THE CURSOR
5584	99D8	C3	4F	00	JMP ZESCND
5585	99D8	.	.	.	;*****
5586	99D8	.	.	.	; NOFUNC--DO NOTHING
5587	99D8	.	.	.	;*****
5588	99D8	.	.	.	NOFUNC EQU \$
5589	99D8	C9	.	.	RET
5590	99DC	.	.	.	;*****
5591	99DC	.	.	.	; IGNSEQ--ESC * <BAD> RECEIVED. IGNORE EVERYTHING
5592	99DC	.	.	.	; UNTIL UPPER CASE LETTER OR ESCAPE RECEIVED
5593	99DC	.	.	.	;*****
5594	99DC	.	.	.	IGNSEQ EQU \$
5595	99DC	21	B0	61	LXI H,IGNTAB ;SET RANGE TABLE
5596	99DF	C3	86	63	JMP SETRTB

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 152
=====
5598     99E2     . . .      ;*****
5599     99E2     . . .      ; GRAPHICS SELF TEST
5600     99E2     . . .      ; AS VECTOR IS DRAWN IN SELF TEST, BIT IN
5601     99E2     . . .      ; MEMORY IS COMPARED TO SAMPLE BIT
5602     99E2     . . .      ; ALL VECTORS ARE DRAWN IN COMPLEMENT MODE
5603     99E2     . . .      ; VECTORS ARE DRAWN AS FOLLOWS--
5604     99E2     . . .      ; 1. CLEAR SCREEN. DRAW UP, STARTING AT 0,0
5605     99E2     . . .      ; MOVING FROM LEFT TO RIGHT
5606     99E2     . . .      ; 2. (SCREEN IS NOW SET) DRAW DOWN, STARTING
5607     99E2     . . .      ; AT 719,359, MOVING RT TO LEFT
5608     99E2     . . .      ; 3. CLEAR SCREEN. DRAW LEFT TO RT, STARTING
5609     99E2     . . .      ; AT 0,0, MOVING FROM BOTTOM TO TOP
5610     99E2     . . .      ; 4. (SCREEN IS NOW SET) DRAW RT TO LEFT,
5611     99E2     . . .      ; STARTING AT 719,359, MOVING TOP TO BOTTOM
5612     99E2     . . .      ; TEST IS REPEATED WITH OPPOSITE SENSE OF
5613     99E2     . . .      ; SCREEN (CLEARED WHERE SET BEFORE, ETC)
5614     99E2     . . .      ; ENTRY--DONT CARE
5615     99E2     . . .      ; EXIT---ALL REGISTERS DESTROYED
5616     99E2     . . .      GTEST EQU $
5617     99E2     FB . .      EI ;NEED TIMER INTERRUPTS
5618     99E3     CD 4F 00     CALL ZESCND ;RESET RANGE TABLES
5619     99E6     CD 88 99     CALL DEFLT2 ;PUT INTO KNOWN STATE
5620     99E9     CD 29 6E     CALL ANVOF1 ;TURN A/N OFF, CRT ON
5621     99EC     3E 08 .      MVI A,10Q ;ENABLE SELFTEST
5622     99EE     32 A6 90     STA STFLAG
5623     99F1     21 00 00     LXI H,0 ;CLEAR ERROR FLAGS
5624     99F4     22 B9 90     SHLD PRMBUF ;USE PRMBUF TO STORE ERRORS
5625     99F7     CD F5 9B     CALL CLRERR ;CLEAR HW ERROR FLAG
5626     99FA     CD 78 6D     CALL GCLR1 ;CLEAR SCREEN
5627     99FD     3E 13 .      MVI A,CLRSMP ;SAMPLE = OFF
5628     99FF     CD E1 9A     CALL VTEST ;TEST WITH VERTICAL VECTORS
5629     9A02     3E 13 .      MVI A,CLRSMP ;SAMPLE = OFF
5630     9A04     CD 41 9B     CALL HTEST ;TEST WITH HORIZNTL VECTORS
5631     9A07     CD FD 9B     CALL FAILCK ;SEE IF TEST FAILED
5632     9A0A     . . .      ; NOW REPEAT WITH SCREEN SET
5633     9A0A     CD 83 6D     CALL GSET1 ;SET THE SCREEN
5634     9A0D     3E 33 .      MVI A,SETSMP ;SAMPLE = ON
5635     9A0F     CD E1 9A     CALL VTEST ;TEST WITH VERTICAL VECTORS
5636     9A12     3E 33 .      MVI A,SETSMP ;SAMPLE = ON
5637     9A14     CD 41 9B     CALL HTEST ;TEST WITH HORIZONTAL VECS
5638     9A17     CD FD 9B     CALL FAILCK ;SEE IF TEST FAILED
5639     9A1A     AF . .      XRA A ;DISABLE SELFTEST
5640     9A1B     32 A6 90     STA STFLAG
5641     9A1E     32 07 89     STA CONTST
5642     9A21     CD 78 6D     CALL GCLR1 ;CLEAR THE SCREEN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 153
5644	9A24	.	.	. ;*****	
5645	9A24	.	.	. ; ZOOM TEST--	
5646	9A24	.	.	. ; ZOOM 16X,8X,4X,2X, WHILE DISPLAYING ZOOM SIZE	
5647	9A24	.	.	. ; IN GRAPHICS MEMORY. CHANGE CHAR SIZE WITH ZOOM	
5648	9A24	.	.	. ; SO THAT ZOOMED CHARS STAY THE SAME SIZE	
5649	9A24	.	.	. ; THEN, DRAW CHECKERBOARD PATTERN AT 1X, ZOOM IT,	
5650	9A24	.	.	. ; AND PAN ACROSS IT.	
5651	9A24	.	.	. ; THIS TESTS ZOOM AND PRESHIFT LOGIC, PATTERN, AND	
5652	9A24	.	.	. ; AND PRESCALE	
5653	9A24	.	.	. ;*****	
5654	9A24	3A	AD 90	LDA TKFLGS ;LOAD TEK STATE	
5655	9A27	F5	.	PUSH PSW ;SAVE IT (TEST TURN TEK OFF)	
5656	9A28	AF	.	XRA A	
5657	9A29	32	AD 90	STA TKFLGS ;TURN TEK MODE OFF	
5658	9A2C	32	97 90	STA GFLGS6 ;TURN G-TEXT OFF	
5659	9A2F	.	.	. ; INITIALIZE FOR TEST	
5660	9A2F	3E	04 .	MVI A,4 ;SET MODE TO JAM PATTERN	
5661	9A31	CD	20 72	CALL SETMD1	
5662	9A34	3E	04 .	MVI A,4 ;SET LORG TO CENTER, MIDDLE	
5663	9A36	CD	59 99	CALL LORG1	
5664	9A39	21	68 01	LXI H,360 ;SET PEN AND ZOOM POSITION	
5665	9A3C	22	DA 90	SHLD XNEW ;TO X = 360	
5666	9A3F	22	B9 90	SHLD PRMBUF	
5667	9A42	21	B4 00	LXI H,180 ;Y = 180	
5668	9A45	22	D8 90	SHLD YNEW	
5669	9A48	22	8B 90	SHLD PRMBUF+2	
5670	9A4B	CD	32 98	CALL CPUPDA ;MOVE PEN	
5671	9A4E	CD	73 6F	CALL ZPOS1 ;SET ZOOM POSITION	
5672	9A51	CD	41 6E	CALL ZON1 ;TURN ZOOM ON	
5673	9A54	.	.	. ; SET ZOOM TO 16X, CHAR SIZE TO 1X	
5674	9A54	01	00 10	LXI B,16*256+0	
5675	9A57	CD	80 9C	CALL ZMTST	
5676	9A5A	.	.	. ; ZOOM TO 8X, CHAR SIZE 2X	
5677	9A5A	01	01 08	LXI B,8*256+1	
5678	9A5D	CD	80 9C	CALL ZMTST	
5679	9A60	.	.	. ; ZOOM TO 4X, CHAR SIZE 4X	
5680	9A60	01	03 04	LXI B,4*256+3	
5681	9A63	CD	80 9C	CALL ZMTST	
5682	9A66	.	.	. ; ZOOM TO 2X, CHAR SIZE 8X	
5683	9A66	01	07 02	LXI B,2*256+7	
5684	9A69	CD	80 9C	CALL ZMTST	
5685	9A6C	.	.	. ; SET UP FOR PAN TEST	
5686	9A6C	CD	5E 6E	CALL ZOFF1 ;TURN ZOOM OFF	
5687	9A6F	CD	78 6D	CALL GCLR1 ;CLEAR THE SCREEN	
5688	9A72	21	28 01	LXI H,296 ;SET PEN TO 296,116	
5689	9A75	22	DA 90	SHLD XNEW ;SET ZOOM TO 296,180	
5690	9A78	22	B9 90	SHLD PRMBUF	
5691	9A7B	21	74 00	LXI H,116	
5692	9A7E	22	D8 90	SHLD YNEW	
5693	9A81	CD	32 98	CALL CPUPDA	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 154
5694	9A84	CD	73	6F	CALL ZPOS1	
5695	9A87	AF	.	.	XRA A	;SET LONG TO LEFT, BOTTOM
5696	9A88	CD	59	99	CALL LORG1	
5697	9A8B	.	.	.		; CHANGE CHAR LEN SO THAT ENTIRE PATTERN REGISTER
5698	9A8B	.	.	.		; WILL BE EXERCISED
5699	9A8B	21	80	FF	LXI H,-8*16	;CHAR SIZE IS 16X
5700	9A8E	22	D8	FB	SHLD CHLEN	
5701	9A91	21	E0	9A	LXI H,CKBPAT+15	;USE SPECIAL PATTERN
5702	9A94	22	81	90	SHLD CHPAT	;FOR CHECKERBOARD
5703	9A97	21	6A	90	LXI H,CNT1	;DRAW 16 PATTERN BYTES
5704	9A9A	36	10	.	MVI M,16	
5705	9A9C	CD	44	98	CALL CHFIL1	;DRAW THE CHECKERBOARD PAT
5706	9A9F	CD	AA	9C	CALL WAIT15	;WAIT FIRST
5707	9AA2	.	.	.		; TURN ZOOM, CURSOR ON
5708	9AA2	CD	D6	6F	CALL TGCON1	;TURN THE CURSOR ON
5709	9AA5	3E	02	.	MVI A,2	;SET ZOOM TO 3X
5710	9AA7	CD	9E	6E	CALL NWSIZE	
5711	9AAA	CD	41	6E	CALL ZON1	;TURN ZOOM ON
5712	9AAD	.	.	.		; PAN ACROSS THE CHECKERBOARD
5713	9AAD	0E	80	.	MVI C,128	;NO. OF DOTS TO PAN
5714	9AAF	.	.	.	GPT010 EQU \$	
5715	9AAF	C5	.	.	PUSH B	;SAVE
5716	9AB0	.	.	.		; TO PAN, UPDATE X COORD OF CURSOR
5717	9AB0	2A	CF	90	LHLD NEWGCX	
5718	9AB3	23	.	.	INX H	
5719	9AB4	22	CF	90	SHLD NEWGCX	
5720	9AB7	3E	80	.	MVI A,NWZOOM	;USE CURSOR AS CENTER OF
5721	9AB9	CD	40	A2	CALL STFLG5	;ZOOM AREA
5722	9ABC	CD	7B	6F	CALL EOFRM	;DO CURSOR ZOOM UPDATES
5723	9ABF	CD	7B	6F	CALL EOFRM	;WAIT 2 FRAMES
5724	9AC2	C1	.	.	POP B	;RECALL PAN COUNT
5725	9AC3	0D	.	.	DCR C	;ALL DONE?
5726	9AC4	F2	AF	9A	JP GPT010	;NO
5727	9AC7	.	.	.		; FINISH SELF TEST
5728	9AC7	F1	.	.	POP PSW	;RECALL TEK FLAGS
5729	9AC8	32	AD	90	STA TKFLGS	
5730	9ACB	CD	85	99	CALL DEFLT1	;RESET DEFAULTS
5731	9ACE	C3	78	6D	JMP GCLR1	;CLEAR THE SCREEN
5732	9AD1	.	.	.	CKBPAT EQU \$	
5733	9AD1	AA	55	AA	DB 252Q,125Q,252Q,125Q,252Q	
5734	9AD6	55	AA	55	DB 125Q,252Q,125Q,252Q,125Q	
5735	9ADB	AA	55	AA	DB 252Q,125Q,252Q,125Q,252Q	
5736	9AE0	55	.	.	DB 125Q	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 155
=====
5738     9AE1      .      .      .      ;*****
5739     9AE1      .      .      .      ; VTEST--DOES SELF TEST WITH VERTICAL VECTORS
5740     9AE1      .      .      .      ; FILLS SCREEN IN COMPLEMENT MODE WHILE TESTING
5741     9AE1      .      .      .      ; MEMORY AGAINST SAMPLE
5742     9AE1      .      .      .      ; ENTRY A = SAMPLE
5743     9AE1      .      .      .      ;*****
5744     9AE1      .      .      .      VTEST EQU $
5745     9AE1      32     B5     90      STA CURMOD      ;STORE SAMPLE
5746     9AE4      .      .      .      ; DRAW FROM BOTTOM UP, MOVING LEFT TO RIGHT
5747     9AE4      21     D0     02      LXI H,720      ;DRAW 720 VERTICAL VECTORS
5748     9AE7      22     51     90      SHLD VECCNT
5749     9AEA      21     00     00      LXI H,0        ;SET STARTING POINT =0,0
5750     9AED      22     DE     90      SHLD XCURR     ;XCURR,YCURR=START OF VEC
5751     9AF0      22     DC     90      SHLD YCURR
5752     9AF3      22     DA     90      SHLD XNEW     ;SET ENDPT = 0,359
5753     9AF6      21     67     01      LXI H,359     ;XNEW,YNEW=ENDPOINT
5754     9AF9      22     D8     90      SHLD YNEW
5755     9AFC      .      .      .      VT010 EQU $
5756     9AFC      CD     A1     9B      CALL STDRAW    ;DRAW/TEST THE VECTOR
5757     9AFF      CA     0F     9B      JZ VT020      ;ALL 720 VECTORS DRAWN?
5758     9B02      2A     DE     90      LHLD XCURR    ;NO MOVE RT ONE POINT
5759     9B05      23     .      .      INX H
5760     9B06      22     DE     90      SHLD XCURR
5761     9B09      22     DA     90      SHLD XNEW
5762     9B0C      C3     FC     9A      JMP VT010     ;GO THRU AGAIN
5763     9B0F      .      .      .      ;*****
5764     9B0F      .      .      .      ; DRAW FROM TOP DOWN, MOVING LEFT TO RIGHT
5765     9B0F      .      .      .      ;*****
5766     9B0F      .      .      .      VT020 EQU $
5767     9B0F      21     D0     02      LXI H,720     ;DRAW 720 VERTICAL VECTORS
5768     9B12      22     51     90      SHLD VECCNT
5769     9B15      2B     .      .      DCX H
5770     9B16      22     DE     90      SHLD XCURR    ;SET START = 719,359
5771     9B19      22     DA     90      SHLD XNEW
5772     9B1C      21     67     01      LXI H,359
5773     9B1F      22     DC     90      SHLD YCURR    ;DRAW VECTORS FROM TOP TO BO
5774     9B22      21     00     00      LXI H,0
5775     9B25      22     D8     90      SHLD YNEW
5776     9B28      3A     B5     90      LDA CURMOD    ;LOAD SAMPLE
5777     9B28      EE     20     .      XRI 40Q      ;COMPLEMENT SAMPLE
5778     9B20      32     B5     90      STA CURMOD
5779     9B30      .      .      .      VT030 EQU $
5780     9B30      CD     A1     9B      CALL STDRAW    ;DRAW DOWN
5781     9B33      C8     .      .      RZ           ;ALL 720 DONE?
5782     9B34      2A     DE     90      LHLD XCURR    ;NOT DONE--MOVE LEFT 1 DOT
5783     9B37      2B     .      .      DCX H
5784     9B38      22     DE     90      SHLD XCURR
5785     9B3B      22     DA     90      SHLD XNEW
5786     9B3E      C3     30     9B      JMP VT030     ;GO THRU AGAIN
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 156
5788	9B41	. . .	;*****	
5789	9B41	. . .	; HTEST--DOES SELF TEST WITH HORIZONTAL VECTORS	
5790	9B41	. . .	; FILLS SCREEN IN COMPLEMENT MODE WHILE TESTING	
5791	9B41	. . .	; MEMORY AGAINST SAMPLE	
5792	9B41	. . .	; ENTRY A = SAMPLE	
5793	9B41	. . .	;*****	
5794	9B41	. . .	HTEST EQU \$	
5795	9B41	32 B5 90	STA CURMOD ;STORE SAMPLE	
5796	9B44	. . .	; DRAW FROM LEFT TO RIGHT, MOVING BOTTOM TO TOP	
5797	9B44	21 68 01	LXI H,360 ;DRAW 360 HORIZ VECTORS	
5798	9B47	22 51 90	SHLD VECCNT	
5799	9B4A	21 00 00	LXI H,0 ;SET START TO 0,0	
5800	9B4D	22 DE 90	SHLD XCURR	
5801	9B50	22 DC 90	SHLD YCURR	
5802	9B53	22 D8 90	SHLD YNEW	
5803	9B56	21 CF 02	LXI H,719 ;ENDPOINT IS 719,0	
5804	9B59	22 DA 90	SHLD XNEW	
5805	9B5C	. . .	HT010 EQU \$	
5806	9B5C	CD A1 9B	CALL STDRAW ;DRAW/TEST THE VECTOR	
5807	9B5F	CA 6F 9B	JZ HT020 ;ALL 360 VECTORS DRAWN?	
5808	9B62	2A DC 90	LHLD YCURR ;NO--MOVE UP ONE	
5809	9B65	23 . .	INX H	
5810	9B66	22 DC 90	SHLD YCURR	
5811	9B69	22 D8 90	SHLD YNEW	
5812	9B6C	C3 5C 9B	JMP HT010	
5813	9B6F	. . .	;*****	
5814	9B6F	. . .	; DRAW FROM RIGHT TO LEFT, MOVING TOP TO BOTTOM	
5815	9B6F	. . .	;*****	
5816	9B6F	. . .	HT020 EQU \$	
5817	9B6F	21 68 01	LXI H,360 ;360 VECTORS AGAIN	
5818	9B72	22 51 90	SHLD VECCNT	
5819	9B75	2B . .	DCX H	
5820	9B76	22 DC 90	SHLD YCURR ;SET START = 719,359	
5821	9B79	22 D8 90	SHLD YNEW ;SET END = 0,359	
5822	9B7C	21 CF 02	LXI H,719	
5823	9B7F	22 DE 90	SHLD XCURR	
5824	9B82	21 00 00	LXI H,0	
5825	9B85	22 DA 90	SHLD XNEW	
5826	9B88	3A B5 90	LDA CURMOD ;COMPLEMENT SAMPLE BIT	
5827	9B88	EE 20 .	XRI 40Q	
5828	9B8D	32 B5 90	STA CURMOD	
5829	9B90	. . .	HT030 EQU \$;DRAW LEFT	
5830	9B90	CD A1 9B	CALL STDRAW ;DRAW/TEST THE VECTOR	
5831	9B93	C8 . .	RZ ;ALL DONE?	
5832	9B94	2A DC 90	LHLD YCURR ;NO--MOVE DOWN ONE	
5833	9B97	2B . .	DCX H	
5834	9B98	22 DC 90	SHLD YCURR	
5835	9B9B	22 D8 90	SHLD YNEW	
5836	9B9E	C3 90 9B	JMP HT030 ;GO THRU AGAIN	

=====					PAGE 157	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS		
=====						
5838	9BA1	.	.	.	;*****	
5839	9BA1	.	.	.	; STDRAW--DOES DRAW/TEST OF A VECTOR FROM	
5840	9BA1	.	.	.	; CURRENT POINT TO NEW POINT	
5841	9BA1	.	.	.	; EXIT Z => ALL VECTORS DRAWN	
5842	9BA1	.	.	.	;*****	
5843	9BA1	.	.	.	STDRAW EQU \$	
5844	9BA1	CD	5D	65	CALL SETUP	;COMPUTE VECTOR PARAMS
5845	9BA4	3E	08	.	MVI A,NEWWA	;MUST USE NEW WA
5846	9BA6	CD	26	A2	CALL STFLG1	
5847	9BA9	3E	08	.	MVI A,DWFRST	;MUST DRAW FIRST DOT
5848	9BAB	CD	40	A2	CALL STFLG5	
5849	9BAE	CD	60	66	CALL DRWVEC	;DRAW THE VECTOR
5850	9BB1	.	.	.	STD010 EQU \$	
5851	9BB1	CD	87	A2	CALL WAIT	;WAIT FOR IDLE HW
5852	9BB4	3A	20	89	LDA HWSTAT	;FETCH SELF TEST STATUS
5853	9BB7	E6	40	.	ANI STBIT	;FAIL BIT SET?
5854	9BB9	C2	CA	9B	JNZ STFAL	;YES--REPORT FAILURE
5855	9BBC	.	.	.	; DISABLE CONTINUE SELF TEST	
5856	9BBC	AF	.	.	XRA A	
5857	9BBD	32	07	89	STA CONTST	;CLEAR CONTINUE SELF TEST
5858	9BC0	2A	51	90	LHLD VECNT	;UPDATE VECTOR COUNT
5859	9BC3	2B	.	.	DCX H	
5860	9BC4	22	51	90	SHLD VECNT	
5861	9BC7	7C	.	.	MOV A,H	;TEST FOR VECNT=0
5862	9BC8	B5	.	.	ORA L	;Z FLAG SET IF DONE
5863	9BC9	C9	.	.	RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 158
=====
5865      9BCA      . . .      ;*****
5866      9BCA      . . .      ; STFAIL -- DETERMINES BAD PACK AND SETS
5867      9BCA      . . .      ; FAILURE BIT FOR IT IN PRMBUF,PRMBUF+1. THERE
5868      9BCA      . . .      ; IS ONE BIT FOR EACH PACK, WHICH IS SET IF
5869      9BCA      . . .      ; THAT PACK FAILS
5870      9BCA      . . .      ;*****
5871      9BCA      . . .      STFAIL EQU $
5872      9BCA      . . .      ;COMPUTE THE BIT NO. FOR THE PACK
5873      9BCA      3A 20 89      LDA  HWSTAT      ;FETCH BAD PACK
5874      9BCD      E6 1E .      ANI  360          ;A = BAD PACK NUMBER*2
5875      9BCF      1F . . .      RAR                      ;A = BAD PACK # (0-15)
5876      9BD0      21 B9 90      LXI  H,PRMBUF    ;ASSUME PACK IS FROM 0-7
5877      9BD3      FE 08 .      CPI  100          ;IS IT REALLY?
5878      9BD5      DA DB 9B      JC   STF010      ;YES
5879      9BD8      23 . . .      INX  H           ;NO, USE PRMBUF+1
5880      9BD9      D6 08 .      SUI  100          ;CONVERT 8-15 TO 0-7
5881      9BD8      . . .      STF010 EQU $
5882      9BD8      . . .      ; SET BIT IN PRMBUF (OR PRMBUF+1) CORRESPONDING
5883      9BD8      . . .      ; TO BAD PACK
5884      9BD8      4F . . .      MOV  C,A         ;C = BIT NUMBER (0-7)
5885      9BDC      AF . . .      XRA  A           ;A = BIT
5886      9BDD      37 . . .      STC                      ;SET CY FOR 1ST BIT POSITION
5887      9BDE      . . .      STF020 EQU $
5888      9BDE      17 . . .      RAL                      ;MOVE BIT LEFT ONE
5889      9BDF      0D . . .      DCR  C           ;BIT IN PROPER POSITION?
5890      9BE0      F2 DE 9B      JP   STF020      ;NO, SHIFT IT LEFT ONE MORE
5891      9BE3      B6 . . .      ORA  M           ;YES, MERGE BIT INTO PRMBUF
5892      9BE4      77 . . .      MOV  M,A         ;STORE UPDATED FAIL FLAGS
5893      9BE5      . . .      ;CONTINUE THE TEST FOR THIS LINE
5894      9BE5      3E 08 .      MVI  A,100
5895      9BE7      32 07 89      STA  CONTST      ;SET CONTINUE SELF TEST
5896      9BEA      CD F5 9B      CALL CLRERR      ;CLEAR THE ERROR FLAG
5897      9BED      3E 01 .      MVI  A,BUSY      ;START DRAWING AGAIN
5898      9BEF      32 20 89      STA  HWFLGS      ;AND CLEAR ST FAIL FLAG
5899      9BF2      C3 B1 9B      JMP  STD010
5900      9BF5      . . .      ;*****
5901      9BF5      . . .      ; CLRERR--CLEAR SELF TEST ERROR FLAG
5902      9BF5      . . .      ;*****
5903      9BF5      . . .      CLRERR EQU $
5904      9BF5      CD 87 A2      CALL WAIT        ;INSURE HW IDLE
5905      9BF8      AF . . .      XRA  A
5906      9BF9      32 20 89      STA  HWFLGS
5907      9BFC      C9 . . .      RET
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 159
5909	9BFD	. . .	;*****	
5910	9BFD	. . .	; FAILCK--SEE IF SELF TEST FAILED. IF SO, STOP	
5911	9BFD	. . .	; THE TEST AND REPORT THE ERROR.	
5912	9BFD	. . .	;*****	
5913	9BFD	. . .	FAILCK EQU \$	
5914	9BFD	2A B9 90	LHLD PRMBUF ;GET FAIL FLAGS	
5915	9C00	7C . .	MOV A,H ;ARE ANY OF THEM SET?	
5916	9C01	B5 . .	ORA L	
5917	9C02	C8 . .	RZ ;NO, CONTINUE THE TEST	
5918	9C03	. . .	; FALL INTO ERROR REPORTING ROUTINE	
5919	9C03	. . .	;*****	
5920	9C03	. . .	; REPORT--REPORT SELF TEST FAIL	
5921	9C03	. . .	; IF TEST INITIATED REMOTELY, RESET THE TERMINAL	
5922	9C03	. . .	; IF LOCAL, HANG THE TERMINAL UNTIL RESET HIT	
5923	9C03	. . .	;*****	
5924	9C03	. . .	REPORT EQU \$	
5925	9C03	CD C3 00	CALL ZDCIO ;INITIATED REMOTELY?	
5926	9C06	C2 00 00	JNZ 0 ;YES, RESET THE TERMINAL	
5927	9C09	CD ED 6D	CALL GVOFF1 ;TURN GRAPHICS OFF	
5928	9C0C	CD 18 6E	CALL ANVON1 ;TURN A/N ON	
5929	9C0F	21 57 9C	LXI H,STFMSG ;PUT UP FAIL MESSAGE	
5930	9C12	22 F1 FF	SHLD ZMSGP1	
5931	9C15	B7 . .	ORA A ;(ADD TO DISPLAY)	
5932	9C16	CD 40 00	CALL ZDSPMG	
5933	9C19	21 72 9C	LXI H,STMSG2 ;PART OF MESSAGE THAT	
5934	9C1C	22 F1 FF	SHLD ZMSGP1 ;DOESNT CHANGE	
5935	9C1F	. . .	; PUT UP MESSAGE FOR FIRST 8 CHIPS	
5936	9C1F	21 0D FB	LXI H,LBLBUF ;MESSAGE BUFFER	
5937	9C22	22 EF FF	SHLD ZMSGP2	
5938	9C25	23 . .	INX H ;SKIP OVER ROW NUMBER	
5939	9C26	36 31 .	MVI M,ONE ;STORE COLUMN NUMBER	
5940	9C28	23 . .	INX H	
5941	9C29	36 CE .	MVI M,ZEOP ;STORE END OF MESSAGE	
5942	9C2B	3A B9 90	LDA PRMBUF ;FAIL BITS PACKS 0-7	
5943	9C2E	CD 3E 9C	CALL RPT020 ;PUT UP MESSAGE	
5944	9C31	. . .	; PUT UP MESSAGE FOR LAST 8 CHIPS	
5945	9C31	21 0E FB	LXI H,LBLBUF+1 ;PTR TO COLUMN	
5946	9C34	34 . .	INR M ;SET TO COLUMN 2	
5947	9C35	3A BA 90	LDA PRMBUF+1 ;FAIL BITS PACKS 8-15	
5948	9C38	CD 3E 9C	CALL RPT020 ;PUT UP MESSAGE	
5949	9C3B	C3 EA 00	JMP ZHANG ;HANG THE TERMINAL	
5950	9C3E	. . .	RPT020 EQU \$	
5951	9C3E	. . .	; DISPLAY MESSAGE FOR BAD CHIPS	
5952	9C3E	. . .	; A = FLAGS FOR BAD CHIPS	
5953	9C3E	. . .	; LBLBUF = POINTER TO BAD ROW NUMBER	
5954	9C3E	. . .	; SET B = 8 FOR 8 ITERATIONS	
5955	9C3E	. . .	; SET C = ASCII 1 FOR PACK VALUE	
5956	9C3E	01 31 08	LXI B,8*256+ONE	
5957	9C41	. . .	RPT030 EQU \$	
5958	9C41	1F . .	RAR ;ROTATE FAIL BIT INTO CY	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 160
=====
5959     9C42     D2  51  9C          JNC  RPT040      ;CHIP IS OK, GET NEXT
5960     9C45     F5  .   .          PUSH PSW        ;BAD CHIP, SAVE A
5961     9C46     C5  .   .          PUSH B          ;SAVE C
5962     9C47     21 0D  FB        LXI  H,LBLBUF   ;STORE ROW NUMBER
5963     9C4A     71  .   .          MOV  M,C
5964     9C4B     B7  .   .          ORA  A          ;(ADD MESSAGE TO DISPLAY)
5965     9C4C     CD 40  00        CALL ZDSPMG
5966     9C4F     C1  .   .          POP  B
5967     9C50     F1  .   .          POP  PSW
5968     9C51     .   .   .          RPT040 EQU $
5969     9C51     0C  .   .          INR  C          ;UPDATE PACK COUNT
5970     9C52     05  .   .          DCR  B          ;ALL 8 DONE ?
5971     9C53     C8  .   .          RZ              ;YES
5972     9C54     C3 41  9C        JMP  RPT030     ;NO, DO THE NEXT
5973     9C57     .   .   .          ;
5974     9C57     .   .   .          STFMSG EQU $
5975     9C57     CC  .   .          DB   EOL
5976     9C58     47 52  41        DB   'GRAPHICS SELF TEST ERROR',EOL
5977     9C71     CE  .   .          DB   ZEOP
5978     9C72     .   .   .          STMSG2 EQU $
5979     9C72     4D 45  4D        DB   'MEMORY CHIP U',0
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 161
=====
5981      9C80      . . .      ;*****
5982      9C80      . . .      ; ZMTST--DO ZOOM TEST
5983      9C80      . . .      ; ENTRY  B = ZOOM SIZE
5984      9C80      . . .      ;          C = CHAR SIZE
5985      9C80      . . .      ; WAIT 1.5 SECOND BEFORE RETURNING
5986      9C80      . . .      ;*****
5987      9C80      . . .      ZMTST EQU $
5988      9C80      C5 . . .      PUSH B          ;SAVE ZOOM SIZE
5989      9C81      79 . . .      MOV A,C
5990      9C82      CD F1 77      CALL TXSIZ1     ;SET TEXT SIZE
5991      9C85      C1 . . .      POP B          ;RECALL ZOOM SIZE
5992      9C86      78 . . .      MOV A,B
5993      9C87      F5 . . .      PUSH PSW       ;SAVE ZOOM SIZE
5994      9C88      21 0D FB      LXI H,LBLBUF   ;MESSAGE BUFFER
5995      9C88      CD AB 00      CALL ZBNDCA    ;CONVERT TO DEC AND STORE
5996      9C8E      36 58 . . .      MVI M,130Q    ;STORE A CAP X
5997      9C90      F1 . . .      POP PSW       ;RECALL ZOOM SIZE
5998      9C91      F5 . . .      PUSH PSW      ;SAVE ZOOM SIZE AGAIN
5999      9C92      3D . . .      DCR A         ;SET NEW ZOOM SIZE
6000      9C93      CD 9E 6E      CALL NWSIZE
6001      9C96      CD 7B 6F      CALL EOFRM     ;DO ZOOM UPDATE
6002      9C99      F1 . . .      POP PSW       ;RECALL ZOOM SIZE
6003      9C9A      0E 02 . . .      MVI C,2       ;ASSUME 2 CHARS
6004      9C9C      FE 0A . . .      CPI 10        ;ZOOM LARGER THAN 9?
6005      9C9E      DA A3 9C      JC ZMT010     ;NO
6006      9CA1      0E 03 . . .      MVI C,3       ;YES, REALLY 3 CHAR
6007      9CA3      . . .      ZMT010 EQU $
6008      9CA3      21 74 90      LXI H,LBLCTR  ;SET LABEL COUNT
6009      9CA6      71 . . .      MOV M,C
6010      9CA7      CD 20 99      CALL SNLBO    ;PRINT THE CHARACTERS
6011      9CAA      . . .      WAIT15 EQU $
6012      9CAA      2E 96 . . .      MVI L,150    ;DO THE DELAY
6013      9CAC      C3 B4 00      JMP ZDELAY    ;FOR 1.5 SECONDS
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 162
=====
6015     9CAF      . . .      ;*****
6016     9CAF      . . .      ; CONTROL CODES--
6017     9CAF      . . .      ; TEST TO SEE IF CONTROL CODE IS TO BE EXECUTED
6018     9CAF      . . .      ; FOR GRAPHICS TEXT.  IF SO, DO NOT EXECUTE
6019     9CAF      . . .      ; IN A/N
6020     9CAF      . . .      ;
6021     9CAF      . . .      ; EXIT  CY=> DONT EXECUTE IN A/N
6022     9CAF      . . .      ;          NC=> EXECUTE IN A/N
6023     9CAF      . . .      ;*****
6024     9CAF      . . .      ;*****
6025     9CAF      . . .      ; XCR--DO A GRAPHICS CARRIAGE RETURN
6026     9CAF      . . .      ; IF IN TEK MODE, MOVE TO PROPER MARGIN
6027     9CAF      . . .      ; ENTRY--DONT CARE
6028     9CAF      . . .      ; EXIT---CY => DO NOT EXECUTE IN A/N
6029     9CAF      . . .      ;          NC => EXECUTE IN A/N
6030     9CAF      . . .      ;          A DESTROYED, HL, DE SAVED
6031     9CAF      . . .      ;*****
6032     9CAF      . . .      XCR      EQU      $
6033     9CAF      CD  CC  9D      CALL  CCCHK      ;SOFT KEYS OR DISP FUNC?
6034     9CB2      D0 . .      RNC          ;YES, PROCESS IN A/N
6035     9CB3      E5 . .      PUSH H      ;NO, SAVE REGISTERS
6036     9CB4      D5 . .      PUSH D      ;SAVE D
6037     9CB5      3A AD  90      LDA  TKFLGS    ;IN TEK MODE?
6038     9CB8      4F . .      MOV  C,A      ;SAVE TEK FLAGS
6039     9CB9      E6 40 .      ANI  SCLD     ;(SCALED ONLY)
6040     9CBB      C2 DD  9C      JNZ  XCR010    ;YES-DO A TEK CR
6041     9CBE      . . .      ; IF LABEL BUFFER HAS ANYTHING IN IT, PRINT IT
6042     9CBE      CD 20 99      CALL  SNLBO
6043     9CC1      CD  F4  78      CALL  PCH050   ;DO CURSOR UPDATES
6044     9CC4      . . .      ;MOVE CURRENT POINT TO START OF LINE
6045     9CC4      . . .      ; ONLY CHANGE X OR Y COORD THAT CORRESPONDS TO
6046     9CC4      . . .      ; MARGIN
6047     9CC4      3A DB  FB      LDA  TANG      ;FETCH CHARACTER DIRECTION
6048     9CC7      0F . .      RRC          ;CHECK LSBIT
6049     9CC8      DA  D4  9C      JC   XCR005
6050     9CCB      . . .      ; ANGLES 0 AND 2 WANT X CHANGED
6051     9CCB      2A 7B  90      LHLD XSOL     ;SET X COORD TO START OF LIN
6052     9CCE      22 DA  90      SHLD XNEW
6053     9CD1      C3  F2  9C      JMP  XCR030    ;UPDATE CURRENT POINT
6054     9CD4      . . .      XCR005 EQU $
6055     9CD4      . . .      ; ANGLES 1 AND 3 WANT Y CHANGED
6056     9CD4      2A 79  90      LHLD YSOL     ;SET Y COORD TO START OF LIN
6057     9CD7      22 D8  90      SHLD YNEW
6058     9CDA      C3  F2  9C      JMP  XCR030
6059     9CDD      . . .      ; DO A TEK CARRIAGE RETURN
6060     9CDD      . . .      XCR010 EQU $
6061     9CDD      21 00 00      LXI  H,XMARG0 ;ASSUME AT MARGIN 0
6062     9CE0      3E 02 .      MVI  A,MARG1  ;REALLY THERE?
6063     9CE2      A1 . .      ANA  C
6064     9CE3      CA  E9  9C      JZ   XCR020   ;YES, LEAVE AS IS
=====

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 163
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
6065	9CE6	21	03	01	LXI H,XMARG1 ;NO, SET MARGIN 1	
6066	9CE9	.	.	.	XCR020 EQU \$	
6067	9CE9	22	DA	90	SHLD XNEW ;STORE NEW X COORD	
6068	9CEC	2A	DC	90	LHLD YCURR ;Y COORD UNCHANGED	
6069	9CEF	22	D8	90	SHLD YNEW	
6070	9CF2	.	.	.	XCR030 EQU \$	
6071	9CF2	CD	85	78	CALL PCH1 ;UPDATE CURRENT POINT	
6072	9CF5	.	.	.	XCR035 EQU \$	
6073	9CF5	D1	.	.	POP D ;RESTORE REGISTERS	
6074	9CF6	E1	.	.	POP H	
6075	9CF7	37	.	.	STC ;CY => DONT PROCESS FURTHER	
6076	9CF8	C9	.	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 164
6078	9CF9	.	.	*****	
6079	9CF9	.	.	; XLF--GRAPHICS LINE FEED	
6080	9CF9	.	.	; ENTRY--DONT CARE	
6081	9CF9	.	.	; EXIT---CY => DO NOT EXECUTE IN A/N	
6082	9CF9	.	.	; NC => EXECUTE IN A/N	
6083	9CF9	.	.	; A DESTROYED, HL, DE SAVED	
6084	9CF9	.	.	*****	
6085	9CF9	.	.	XLF EQU \$	
6086	9CF9	CD	CC 9D	CALL CCCHK ;DISP FUNCS OR SOFT KEYS?	
6087	9CFC	D0	.	RNC ;YES, PROCESS IN A/N	
6088	9CFD	E5	.	PUSH H ;NO, SAVE H	
6089	9CFE	D5	.	PUSH D ;SAVE D	
6090	9CFF	.	.	; IF LABEL BUFFER NOT EMPTY, SEND IT	
6091	9CFF	CD	20 99	CALL SNDLBO	
6092	9D02	.	.	XLF010 EQU \$	
6093	9D02	CD	F4 78	CALL PCH050 ;DO CURSOR UPDATES	
6094	9D05	2A	85 90	LHLD XLFINC ;ADD LINE FEED SPACING	
6095	9D08	EB	.	XCHG	
6096	9D09	2A	83 90	LHLD YLFINC	
6097	9D0C	CD	55 9D	CALL LFUPDA ;TO CURRENT POINT	
6098	9D0F	.	.	; IF IN SCALED TEK MODE, AND LF WAS OFF THE	
6099	9D0F	.	.	; SCREEN, HOME UP	
6100	9D0F	3A	AD 90	LDA TKFLGS ;IN SCALED TEK MODE?	
6101	9D12	4F	.	MOV C,A ;(SAVE FLAGS)	
6102	9D13	E6	40 .	ANI SCLD	
6103	9D15	CA	F2 9C	JZ XCR030 ;NO, EXIT	
6104	9D18	.	.	; SEE IF LINE FEED WENT NEGATIVE	
6105	9D18	7C	.	MOV A,H	
6106	9D19	B7	.	ORA A ;OFF THE SCREEN?	
6107	9D1A	F2	F2 9C	JP XCR030 ;NO, EXIT	
6108	9D1D	3E	02 .	MVI A,MARG1 ;TOGGLE MARGIN 1 FLAG	
6109	9D1F	A9	.	XRA C	
6110	9D20	32	AD 90	STA TKFLGS ;STORE NEW TEK FLAGS	
6111	9D23	CD	50 6A	CALL TEKHOM ;HOME THE CURSOR	
6112	9D26	.	.	; SEND BREAK IF PF BREAK STRAP IN	
6113	9D26	3A	C4 FB	LDA TEKPF ;FETCH PAGE FULL STRAPS	
6114	9D29	E6	01 .	ANI PFBRAK ;SEND BREAK?	
6115	9D2B	CA	39 9D	JZ XLF020 ;NO	
6116	9D2E	.	.	; SEND BREAK	
6117	9D2E	CD	D2 00	CALL ZCKRMT ;IN REMOTE?	
6118	9D31	CA	39 9D	JZ XLF020 ;NO, DONT SEND BREAK	
6119	9D34	3E	05 .	MVI A,PUTBRK	
6120	9D36	CD	11 50	CALL ZDCCTL ;SEND THE BREAK SIGNAL	
6121	9D39	.	.	XLF020 EQU \$	
6122	9D39	.	.	; HANG TERMINAL IF PF BUSY FLAG SET?	
6123	9D39	3A	C4 FB	LDA TEKPF	
6124	9D3C	E6	02 .	ANI PFBUSY ;BUSY?	
6125	9D3E	CA	F5 9C	JZ XCR035 ;NO, EXIT	
6126	9D41	.	.	; HANG THE TERMINAL UNTIL CLEAR KEY HIT	
6127	9D41	.	.	TKBUSY EQU \$	

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 165
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 165
6128	9D41	CD 28	A4	CALL GETKEY ;GET A KEY	
6129	9D44	C2 41	9D	JNZ TKBUSY ;NO KEY, WAIT	
6130	9D47	FE 8D	.	CPI CLRKEY ;IS IT THE CLEAR KEY?	
6131	9D49	C2 41	9D	JNZ TKBUSY ;NO, LOOP	
6132	9D4C	CD 1D	6A	CALL PAGE ;YES, DO THE 'PAGE'	
6133	9D4F	C3 F5	9C	JMP XCR035 ;EXIT	
6134	9D52	C3 F5	9C	JMP XCR035 ;EXIT	

```
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 166
=====
6136     9D55      . . .      ;*****
6137     9D55      . . .      ; LFUPDA--ADD INCREMENT TO CURRENT POINT
6138     9D55      . . .      ; DONT UPDATE SOL IF INCREMENT IS 0
6139     9D55      . . .      ; ENTRY DE = XINC, HL = Y INC
6140     9D55      . . .      ; EXIT HL = NEW Y COORD
6141     9D55      . . .      ;*****
6142     9D55      . . .      LFUPDA EQU $
6143     9D55     E5 . .      PUSH H ;SAVE Y INC
6144     9D56     2A DE 90    LHLD XCURR ;UPDATE X COORD FIRST
6145     9D59     7B . .      MOV A,E ;IS X INCREMENT = 0?
6146     9D5A     B2 . .      ORA D
6147     9D5B     CA 62 9D    JZ LFD010 ;YES, DONT UPDATE SOL
6148     9D5E     19 . .      DAD D ;ADD INCREMENT TO X
6149     9D5F     22 7B 90    SHLD XSOL ;UPDATE START OF LINE
6150     9D62      . . .      LFD010 EQU $
6151     9D62     22 DA 90    SHLD XNEW ;UPDATE X
6152     9D65     D1 . .      POP D ;RECALL Y INCREMENT
6153     9D66     2A DC 90    LHLD YCURR ;DE = INC, HL = CURRENT Y
6154     9D69     7B . .      MOV A,E ;IS INCREMENT = 0?
6155     9D6A     B2 . .      ORA D
6156     9D6B     CA 72 9D    JZ LFD020 ;YES, DONT UPDATE SOL
6157     9D6E     19 . .      DAD D ;ADD INCREMENT TO Y
6158     9D6F     22 79 90    SHLD YSOL ;UPDATE SOL
6159     9D72      . . .      LFD020 EQU $
6160     9D72     22 D8 90    SHLD YNEW
6161     9D75     C9 . .      RET
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 167
6163	9D76	.	.	. ;*****	
6164	9D76	.	.	. ; XBS--GRAPHICS BACKSPACE	
6165	9D76	.	.	. ; ENTRY--DONT CARE	
6166	9D76	.	.	. ; EXIT---ALL REGISTERS DESTROYED	
6167	9D76	.	.	. ; CY => DO NOT EXECUTE IN A/N	
6168	9D76	.	.	. ; NC => EXECUTE IN A/N	
6169	9D76	.	.	. ;*****	
6170	9D76	.	.	. XBS EQU \$	
6171	9D76	CD	CC 9D	CALL CCCHK ;DISP FUNCS OR SOFT KEYS?	
6172	9D79	D0	.	RNC ;YES, EXECUTE IN A/N	
6173	9D7A	CD	F4 78	CALL PCH050 ;DO CURSOR UPDATES	
6174	9D7D	2A	89 90	LHLD XCHINC ;SUBTRACT CHAR SPACING	
6175	9D80	CD	09 A3	CALL NEGATE	
6176	9D83	EB	.	XCHG	
6177	9D84	2A	DE 90	LHLD XCURR	
6178	9D87	19	.	DAD D ;NEW X COORD	
6179	9D88	22	DA 90	SHLD XNEW	
6180	9D8B	2A	87 90	LHLD YCHINC	
6181	9D8E	CD	09 A3	CALL NEGATE	
6182	9D91	EB	.	XCHG	
6183	9D92	2A	DC 90	LHLD YCURR	
6184	9D95	19	.	DAD D	
6185	9D96	22	D8 90	SHLD YNEW ;NEW Y COORD	
6186	9D99	CD	85 78	CALL PCH1 ;UPDATE CURRENT POINT	
6187	9D9C	37	.	STC ;CY => DONT PROCESS FURTHER	
6188	9D9D	C9	.	RET	
6189	9D9E	.	.	. ;*****	
6190	9D9E	.	.	. ; XHT--DO GRAPHICS TAB--SPACE ONE CHAR	
6191	9D9E	.	.	. ; ENTRY--DONT CARE	
6192	9D9E	.	.	. ; EXIT---ALL REGISTERS DESTROYED	
6193	9D9E	.	.	. ; CY => DO NOT EXECUTE IN A/N	
6194	9D9E	.	.	. ; NC => EXECUTE IN A/N	
6195	9D9E	.	.	. ;*****	
6196	9D9E	.	.	. XHT EQU \$	
6197	9D9E	CD	CC 9D	CALL CCCHK ;DISP FUNCS OR SOFT KEYS?	
6198	9DA1	D0	.	RNC ;YES, EXECUTE IN A/N	
6199	9DA2	CD	F4 78	CALL PCH050 ;DO CURSOR UPDATES	
6200	9DA5	CD	6F 78	CALL PCH010 ;DO THE TAB	
6201	9DA8	37	.	STC ;CY=>DONT PROCESS ANY FURTHER	
6202	9DA9	C9	.	RET	
6203	9DAA	.	.	. ;*****	
6204	9DAA	.	.	. ; XVT--GRAPHICS VERTICAL TAB	
6205	9DAA	.	.	. ; ENTRY--DONT CARE	
6206	9DAA	.	.	. ; EXIT---ALL REGISTERS DESTROYED	
6207	9DAA	.	.	. ; CY => DO NOT EXECUTE IN A/N	
6208	9DAA	.	.	. ; NC => EXECUTE IN A/N	
6209	9DAA	.	.	. ;*****	
6210	9DAA	.	.	. XVT EQU \$	
6211	9DAA	CD	CC 9D	CALL CCCHK ;DISP FUNCS OR SOFT KEYS?	
6212	9DAD	D0	.	RNC ;YES, EXECUTE IN A/N	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 168
=====
6213     9DAE     CD  F4  78      CALL PCH050      ;DO CURSOR UPDATES
6214     9DB1     2A  85  90      LHLD XLFINC      ;SUBTRACT LF SPACING
6215     9DB4     CD  09  A3      CALL NEGATE
6216     9DB7     EB  .   .       XCHG
6217     9DB8     2A  83  90      LHLD YLFINC
6218     9DBB     CD  09  A3      CALL NEGATE
6219     9DBE     CD  55  9D      CALL LFUPDA      ;ADD INCREMENT TO CUR. POINT
6220     9DC1     CD  85  78      CALL PCH1        ;UPDATE CURRENT POINT
6221     9DC4     37  .   .       STC              ;CY=>DONT PROCESS ANY FURTHER
6222     9DC5     C9  .   .       RET
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 169
6224	9DC6	.	.	;*****	
6225	9DC6	.	.	; TXCHK--CHECK TO SEE IF IN GRAPHICS TEXT MODE	
6226	9DC6	.	.	; EXIT Z => NOT IN GRAPHICS TEXT MODE	
6227	9DC6	.	.	;*****	
6228	9DC6	.	.	TXCHK EQU \$	
6229	9DC6	3A	97 90	LDA GFLGS6 ;IN GRAPHICS TEXT MODE?	
6230	9DC9	E6	82 .	ANI GTEXT+LABEL	
6231	9DCB	C9	.	RET	
6232	9DCC	.	.	;*****	
6233	9DCC	.	.	; CCCHK--CHECK FOR CONTROL CODE TO BE EXECUTED	
6234	9DCC	.	.	; IN A/N MEMORY. DONE SO IF SOFT KEYS ARE UP,	
6235	9DCC	.	.	; OR IN DISPLAY FUNCTIONS	
6236	9DCC	.	.	; ECHO SUPPRESSION IS CLEARED, TOO	
6237	9DCC	.	.	; EXIT NC => EXECUTE IN A/N MEMORY	
6238	9DCC	.	.	; A DESTROYED	
6239	9DCC	.	.	;*****	
6240	9DCC	.	.	CCCHK EQU \$	
6241	9DCC	CD	A9 6B	CALL CLRSP ;CLEAR ECHO SUPPRESS	
6242	9DCF	CD	C6 00	CALL ZCHKSF ;SOFT KEYS UP?	
6243	9DD2	C0	.	RNZ ;YES, RETURN NC	
6244	9DD3	CD	D9 9D	CALL DFCHK ;IN DISPLAY FUNCTIONS?	
6245	9DD6	C0	.	RNZ ;YES, RETURN NC	
6246	9DD7	37	.	STC ;NO, RETURN CY	
6247	9DD8	C9	.	RET	
6248	9DD9	.	.	;*****	
6249	9DD9	.	.	; DFCHK--SEE IF IN DISPLAY FUNCTIONS	
6250	9DD9	.	.	; EXIT NZ => IN DISPLAY FUNCTIONS	
6251	9DD9	.	.	;*****	
6252	9DD9	.	.	DFCHK EQU \$	
6253	9DD9	3A	F4 FF	LDA ZMDFL1 ;CHECK FLAG IN MDFLG1	
6254	9DDC	E6	01 .	ANI DSPFNC	
6255	9DDE	C9	.	RET	
6256	9DDF	.	.	;*****	
6257	9DDF	.	.	; LEDCHK--SEE IF ALL LEDS ARE ON (SELF TEST IN	
6258	9DDF	.	.	; PROGRESS)	
6259	9DDF	.	.	; EXIT Z => ALL LEDS ARE ON	
6260	9DDF	.	.	;*****	
6261	9DDF	.	.	LEDCHK EQU \$	
6262	9DDF	3A	0C FF	LDA ZKBLED ;LOAD THE LED FLAGS	
6263	9DE2	FE	7F .	CPI 177Q ;ALL ON?	
6264	9DE4	C9	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 170
6266	9DE5	.	.	*****	
6267	9DE5	.	.	; ANCUR--MOVE GRAPHICS CURSOR WITH A/N KEYS IF	
6268	9DE5	.	.	; IN GRAPHICS TEXT MODE	
6269	9DE5	.	.	; ENTRY A = DIRECTION (0-3)	
6270	9DE5	.	.	; EXIT CY => DONT PROCESS ANY FURTHER	
6271	9DE5	.	.	; EXIT---HL,DE,A SAVED	
6272	9DE5	.	.	*****	
6273	9DE5	.	.	ANCUR EQU \$	
6274	9DE5	F5	.	PUSH PSW ;SAVE DIRECTION	
6275	9DE6	CD	C6 9D	CALL TXCHK ;IN GRAPHICS TEXT?	
6276	9DE9	CA	11 9E	JZ AGC010 ;NO, PROCESS IN A/N	
6277	9DEC	CD	CC 9D	CALL CCCHK ;DISPLAY FUNCS OR SOFT KEYS?	
6278	9DEF	D2	11 9E	JNC AGC010 ;YES, PROCESS IN A/N	
6279	9DF2	F1	.	POP PSW ;RECALL DIRECTION	
6280	9DF3	.	.	; DO CURSOR UPDATE IF ITS ON	
6281	9DF3	E5	.	PUSH H ;SAVE REGISTERS	
6282	9DF4	D5	.	PUSH D	
6283	9DF5	F5	.	PUSH PSW ;SAVE A AGAIN	
6284	9DF6	CD	F4 78	CALL PCH050	
6285	9DF9	F1	.	POP PSW ;RECALL A	
6286	9DFA	.	.	; COMPUTE WHICH DIRECTION TO MOVE FROM CURSOR KEY	
6287	9DFA	.	.	; (A REG) AND CURRENT TEXT ANGLE	
6288	9DFA	21	DB FB	LXI H,TANG ;GET TEXT ANGLE	
6289	9DFD	86	.	ADD M ;ADD TO CURSOR KEY	
6290	9DFE	E6	03 .	ANI 30 ;WANT 2 LSB ONLY	
6291	9E00	87	.	ADD A ;MULTIPLY BY 2	
6292	9E01	5F	.	MOV E,A ;SET DE = INDEX TO TABLE	
6293	9E02	16	00 .	MVI D,0	
6294	9E04	21	14 9E	LXI H,ANCTAB ;BASE OF TABLE	
6295	9E07	19	.	DAD D ;POINTER TO CURSOR ROUTINE	
6296	9E08	5E	.	MOV E,M ;FETCH ADDRESS	
6297	9E09	23	.	INX H	
6298	9E0A	56	.	MOV D,M	
6299	9E0B	EB	.	XCHG ;HL = ADDRESS OF ROUTINE	
6300	9E0C	CF	.	RST 1 ;CALL THE ROUTINE	
6301	9E0D	D1	.	POP D ;RESTORE THE REGISTERS	
6302	9E0E	E1	.	POP H	
6303	9E0F	37	.	STC ;RETURN CY	
6304	9E10	C9	.	RET	
6305	9E11	.	.	AGC010 EQU \$	
6306	9E11	F1	.	POP PSW ;RECALL A	
6307	9E12	B7	.	ORA A ;NC => PROCESS IN A/N	
6308	9E13	C9	.	RET	
6309	9E14	.	.	;	
6310	9E14	.	.	ANCTAB EQU \$	
6311	9E14	9E	9D .	DW XHT ;MOVE RIGHT	
6312	9E16	1C	9E .	DW XLF1 ;MOVE DOWN	
6313	9E18	76	9D .	DW XBS ;MOVE LEFT	
6314	9E1A	AA	9D .	DW XVT ;MOVE UP	
6315	9E1C	.	.	;	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 171
=====
6316     9E1C      . . .      ;*****
6317     9E1C      . . .      ; XLF1--SAME AS XLF1 EXCEPT LABEL BUFFER ISNT
6318     9E1C      . . .      ; FLUSHED
6319     9E1C      . . .      ;*****
6320     9E1C      . . .      XLF1 EQU $
6321     9E1C      E5 . .      PUSH H           ;SAVE SAME REGISTERS AS XLF
6322     9E1D      D5 . .      PUSH D
6323     9E1E      C3 02 9D    JMP XLF010       ;THE REST IS THE SAME
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 172
6325	9E21	.	.	*****	
6326	9E21	.	.	; GGTEST--TEST FOR GRAPHICS DATA GET	
6327	9E21	.	.	; RETURNS NZ IF	
6328	9E21	.	.	; 1. AUTO PLOT MENU UP (GET MENU DATA)	
6329	9E21	.	.	; 2. GETTING GRAPHICS IMAGE DATA	
6330	9E21	.	.	; ENTRY--DONT CARE	
6331	9E21	.	.	; EXIT---A DESTROYED	
6332	9E21	.	.	; NZ => GET GRAPHICS DATA	
6333	9E21	.	.	*****	
6334	9E21	.	.	GGTEST EQU \$	
6335	9E21	C3	EE B8	JMP MUCHK ;TEST AUTO PLOT MENU ONLY	
6336	9E24	.	.	*****	
6337	9E24	.	.	; GGINIT--INITIALIZE FOR GRAPHICS DATA GET	
6338	9E24	.	.	; EITHER AUTO PLOT DATA OR IMAGE DATA	
6339	9E24	.	.	; ENTRY--DONT CARE	
6340	9E24	.	.	; EXIT Z => CHAR AVAILABLE	
6341	9E24	.	.	; NZ => NO DATA	
6342	9E24	.	.	; EXIT---ALL REGISTERS DESTROYED	
6343	9E24	.	.	*****	
6344	9E24	.	.	GGINIT EQU \$	
6345	9E24	AF	.	XRA A ;CLEAR GET FLAGS	
6346	9E25	32	C8 FB	STA GGFLGS	
6347	9E28	21	DD FA	LXI H,PTR1 ;SET INIT FLAG FOR PRINTER	
6348	9E2B	3E	01 .	MVI A,PINIT	
6349	9E2D	B6	.	ORA M	
6350	9E2E	77	.	MOV M,A	
6351	9E2F	.	.	; INITIALIZE FOR AUTO PLOT GET	
6352	9E2F	CD	9D 6C	CALL LBLOFF ;TURN LABELS OFF	
6353	9E32	CD	66 9E	CALL FILBF1 ;FILL BUFFER WITH FIRST LINE	
6354	9E35	AF	.	XRA A ;SET Z => DATA	
6355	9E36	C9	.	RET	
6356	9E37	.	.	*****	
6357	9E37	.	.	; GRGET--GET GRAPHICS DATA FROM EITHER MENU OR	
6358	9E37	.	.	; IMAGE MEMORY	
6359	9E37	.	.	; ENTRY--DONT CARE	
6360	9E37	.	.	; EXIT NC = CHAR FOUND, A = CHAR	
6361	9E37	.	.	; CY = NO CHAR	
6362	9E37	.	.	; M = END OF DISPLAY	
6363	9E37	.	.	; Z = END OF FIELD	
6364	9E37	.	.	; P,NZ = END OF LINE	
6365	9E37	.	.	; EXIT---ALL REGISTERS DESTROYED	
6366	9E37	.	.	*****	
6367	9E37	.	.	GRGET EQU \$	
6368	9E37	.	.	; GET AUTO PLOT DATA	
6369	9E37	3A	C8 FB	LDA GGFLGS ;END OF DISPLAY?	
6370	9E3A	E6	01 .	ANI EOD	
6371	9E3C	C2	4C 9E	JNZ GGT010 ;YES, RETURN NO MORE DATA	
6372	9E3F	2A	C5 FB	LHLD GETPTR ;ANYTHING IN BUFFER?	
6373	9E42	7E	.	MOV A,M	
6374	9E43	B7	.	ORA A ;BUFFER EMPTY?	

=====						PAGE 173	
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS		
=====						=====	
6375	9E44	FA	50	9E	JM	FILBUF	;YES, FILL BUFFER UP AGAIN
6376	9E47	23	.	.	INX	H	;NO, UPDATE POINTER
6377	9E48	22	C5	FB	SHLD	GETPTR	
6378	9E4B	C9	.	.	RET		;A = CHAR, NO CARRY
6379	9E4C	.	.	.	GGT010	EQU	\$
6380	9E4C	AF	.	.	XRA	A	;END OF DISPLAY, RETURN
6381	9E4D	3D	.	.	DCR	A	;M, CY
6382	9E4E	37	.	.	STC		
6383	9E4F	C9	.	.	RET		

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 174
=====
6385     9E50      . . .      ;*****
6386     9E50      . . .      ; FILBUF--FILL GET BUFFER WITH MENU DATA
6387     9E50      . . .      ;*****
6388     9E50      . . .      FILBUF EQU $
6389     9E50      3A 02 FB    LDA MUFLO      ;MOVE TO NEXT FIELD
6390     9E53      3C . .     INR A          ;PAST END OF MENU?
6391     9E54      FE 10 .     CPI BOTFLD+1
6392     9E56      DA 63 9E    JC FLB020      ;NO, XFER FIELD TO BUFFER
6393     9E59      . . .      ;AUTO PLOT MENU DONE, SET FLAGS AND EXIT
6394     9E59      . . .      FLB010 EQU $
6395     9E59      21 C8 FB    LXI H,GGFLGS  ;SET END OF DISPLAY
6396     9E5C      3E 01 .     MVI A,EOD     ;FLAG
6397     9E5E      B6 . .     ORA M
6398     9E5F      77 . .     MOV M,A
6399     9E60      C3 86 9E    JMP FLB030     ;RETURN END OF LINE
6400     9E63      . . .      FLB020 EQU $
6401     9E63      CD ED AD    CALL PUTFLD   ;PUT CURSOR IN NEW FIELD
6402     9E66      . . .      FILBF1 EQU $  ;ENTRY FROM INIT ROUTINE
6403     9E66      . . .      ; PUT ESC * A INTO BUFFER
6404     9E66      21 49 FB    LXI H,NUMBUF+60
6405     9E69      22 C5 FB    SHLD GETPTR   ;POINTER TO START OF BUFFER
6406     9E6C      36 1B .     MVI M,330     ;STORE ESC
6407     9E6E      23 . .     INX H
6408     9E6F      36 2A .     MVI M,520     ;STORE *
6409     9E71      23 . .     INX H
6410     9E72      36 61 .     MVI M,1410    ;STORE LOWER CASE A
6411     9E74      23 . .     INX H
6412     9E75      . . .      ; XFER DATA FROM MENU FIELD TO BUFFER
6413     9E75      EB . .     XCHG          ;DE = BUFFER POINTER
6414     9E76      2A FC FA    LHLD DSPFLD   ;HL = LOCATION IN MENU
6415     9E79      CD 33 AB    CALL XFER2     ;PUT INTO GET BUFFER
6416     9E7C      . . .      ; STORE TERMINATOR FOR ESC SEQ
6417     9E7C      EB . .     XCHG          ;HL = BUFFER POINTER
6418     9E7D      3A 02 FB    LDA MUFLO     ;CURRENT FIELD
6419     9E80      C6 48 .     ADI 1100      ;CONVERT TO H-W
6420     9E82      77 . .     MOV M,A       ;STORE CAP LETTER
6421     9E83      23 . .     INX H
6422     9E84      36 CC .     MVI M,EOL     ;STORE END OF DISPLAY
6423     9E86      . . .      ; SET FLAGS TO RETURN END OF LINE
6424     9E86      . . .      FLB030 EQU $
6425     9E86      AF . .     XRA A         ;SET C, P NZ
6426     9E87      3C . .     INR A
6427     9E88      37 . .     STC
6428     9E89      C9 . .     RET
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
6430	9E8A	.	.	.	;*****	175
6431	9E8A	.	.	.	; CURSOR ROUTINES	
6432	9E8A	.	.	.	;*****	
6433	9E8A	.	.	.	;	
6434	9E8A	.	.	.	;*****	
6435	9E8A	.	.	.	; SUPRGC--SUPPRESS THE CURSOR	
6436	9E8A	.	.	.	; ENTRY A = SUPPRESS BIT	
6437	9E8A	.	.	.	;*****	
6438	9E8A	.	.	.	SUPRGC EQU \$	
6439	9E8A	SF	.	.	MOV E,A ;SAVE SUPPRESS BIT	
6440	9E8B	21	AE	90	LXI H,GFLGS5 ;SEE IF RB LINE IS BEING	
6441	9E8E	3E	10	.	MVI A,RBDRW ;DRAWN	
6442	9E90	A6	.	.	ANA M	
6443	9E91	C2	87	A2	JNZ WAIT ;YES-DONT SUPPRESS, JUST WAIT	
6444	9E94	7B	.	.	MOV A,E ;RESTORE SUPPRESS BITS	
6445	9E95	21	80	90	LXI H,GFLGS3	
6446	9E98	B6	.	.	ORA M ;SET THEM	
6447	9E99	77	.	.	MOV M,A	
6448	9E9A	CD	7E	70	CALL RBOFF ;TURN THE RBLINE OFF	
6449	9E9D	21	EC	FB	LXI H,SUPTMR ;RESTART THE SUPPRESS TIMER	
6450	9EA0	36	14	.	MVI M,TIMOUT	
6451	9EA2	CD	DE	9E	CALL GRCOFF ;TURN THE CURSOR OFF	
6452	9EA5	CD	74	A2	CALL SNDGCF ;SEND GC FLAGS TO DRAW	
6453	9EA8	C3	87	A2	JMP WAIT ;WAIT FOR CURSOR TO FINISH	
6454	9EAB	.	.	.	;*****	
6455	9EAB	.	.	.	; ENAB0--CLEAR SUPRO FLAG ONLY IF THE	
6456	9EAB	.	.	.	; AUTO PLOT MENU IS OFF	
6457	9EAB	.	.	.	;*****	
6458	9EAB	.	.	.	ENAB0 EQU \$	
6459	9EAB	CD	EE	B8	CALL MUCHK ;IS THE MENU UP?	
6460	9EAE	C0	.	.	RNZ ;YES, LEAVE THE CURSOR OFF	
6461	9EAF	CD	C6	00	CALL ZCHKSF ;SOFT KEYS UP?	
6462	9EB2	C0	.	.	RNZ ;YES, LEAVE CURSOR OFF	
6463	9EB3	3E	01	.	MVI A,SUPRO ;NO, CLEAR SUPPRESS FLAG	
6464	9EB5	.	.	.	;*****	
6465	9EB5	.	.	.	; ENABGC--UNSUPPRESS THE GRAPHICS CURSOR	
6466	9EB5	.	.	.	; ENTRY A = SUPPRESS BIT	
6467	9EB5	.	.	.	;*****	
6468	9EB5	.	.	.	ENABGC EQU \$	
6469	9EB5	C3	39	A2	JMP CLFLG3 ;CLEAR THE SUPPRESS BIT	
6470	9EB8	.	.	.	; CURSOR TURNED BACK ON IN VR SCAN	
6471	9EB8	.	.	.	;*****	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 176
=====
6473     9EB8      . . .      ; GRCON--TURN GRAPHICS CURSOR ON
6474     9EB8      . . .      ; ENTRY FOR GRCON1  C = GFLGS3
6475     9EB8      . . .      ;*****
6476     9EB8      . . .      GRCON EQU $
6477     9EB8      CD 87 A2    CALL WAIT          ;WAIT FOR IDLE HARDWARE
6478     9EB8      3A B0 90    LDA GFLGS3        ;IS CURSOR ALREADY ON?
6479     9EBE      4F . .     MOV C,A           ;SAVE CURSOR FLAGS
6480     9EBF      E6 40 .     ANI GCON          ;IS CURSOR ACTUALLY ON?
6481     9EC1      C0 . .     RNZ               ;YES--DONE
6482     9EC2      . . .      GRCON1 EQU $
6483     9EC2      3E 0F .     MVI A,SUPRO+SUPR1+SUPR2+TIMSUP
6484     9EC4      A1 . .     ANA C             ;IS CURSOR BEING SUPRESSED?
6485     9EC5      C0 . .     RNZ               ;YES--DONT TURN IT ON
6486     9EC6      . . .      ; TURN CURSOR ON AT NEW CURSOR LOCATION
6487     9EC6      CD 20 A2    CALL NORST        ;DISALLOW RESETS
6488     9EC9      3E 40 .     MVI A,GCON        ;SET 'ON' FLAG
6489     9ECB      CD 33 A2    CALL STFLG3
6490     9ECE      2A CF 90    LHLD NEWGCX      ;UPDATE CURSOR POSITION
6491     9ED1      22 CB 90    SHLD CURGCX
6492     9ED4      EB . .     XCHG              ;ENTRY FOR DRAWGC  DE = X
6493     9ED5      2A CD 90    LHLD NEWGCY      ;HL = Y
6494     9ED8      22 C9 90    SHLD CURGCY
6495     9EDB      C3 FA 9E    JMP DRAWGC        ;DRAW THE CURSOR
6496     9EDE      . . .      ;*****
6497     9EDE      . . .      ; GRCOFF--TURN GRAPHICS CURSOR OFF
6498     9EDE      . . .      ; ENTRY FOR GRCOF1  C = GFLGS3
6499     9EDE      . . .      ;*****
6500     9EDE      . . .      GRCOFF EQU $
6501     9EDE      CD 87 A2    CALL WAIT          ;WAIT FOR IDLE HARD WARE
6502     9EE1      3A B0 90    LDA GFLGS3        ;IS CURSOR ALREADY OFF?
6503     9EE4      4F . .     MOV C,A           ;SAVE CURSOR FLAGS
6504     9EE5      E6 40 .     ANI GCON          ;CURSOR OFF?
6505     9EE7      C8 . .     RZ                ;YES--DONE
6506     9EE8      . . .      GRCOF1 EQU $
6507     9EE8      . . .      ;TURN CURSOR OFF AT CURRENT LOCATION
6508     9EE8      CD 20 A2    CALL NORST        ;DISALLOW RESETS
6509     9EE8      3E 40 .     MVI A,GCON        ;CLEAR 'ON' FLAG
6510     9EED      CD 39 A2    CALL CLFLG3
6511     9EF0      2A CB 90    LHLD CURGCX      ;ENTRY TO DRAWGC
6512     9EF3      EB . .     XCHG              ;DE = X COORD
6513     9EF4      2A C9 90    LHLD CURGCY      ;HL = Y
6514     9EF7      C3 FA 9E    JMP DRAWGC

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 177
6516	9EFA	. . .	;*****	
6517	9EFA	. . .	; DRAWGC--DRAW GRAPHICS CURSOR AT X,Y	
6518	9EFA	. . .	; ENTRY DE = X	
6519	9EFA	. . .	; HL = Y	
6520	9EFA	. . .	;*****	
6521	9EFA	. . .	DRAWGC EQU \$	
6522	9EFA	22 5F 90	SHLD GCY ;SAVE Y COORD	
6523	9EFD	EB . .	XCHG	
6524	9EFE	22 61 90	SHLD GCX ;SAVE X COORD	
6525	9F01	EB . .	XCHG	
6526	9F02	. . .	; COMPUTE STARTING OINT AND LENGTH OF HORIZONTAL	
6527	9F02	. . .	; VECTOR. FIRST COMPUTE THE LEFTMOST POINT	
6528	9F02	21 EC FF	LXI H,MLEN ;- CURSOR LENGTH	
6529	9F05	19 . .	DAD D ;HL = X-LENGTH	
6530	9F06	. . .	;SEE IF LESS THAN 0	
6531	9F06	7C . .	MOV A,H ;CHECK SIGN BIT	
6532	9F07	B7 . .	ORA A	
6533	9F08	F2 0E 9F	JP DGC010 ;IS +	
6534	9F0B	21 00 00	LXI H,0 ;IS -, SET LEFTMOST = 0	
6535	9F0E	. . .	DGC010 EQU \$	
6536	9F0E	22 69 90	SHLD XLEFT ;STORE LEFTMOST POINT	
6537	9F11	. . .	;COMPUTE RIGHTMOST POINT AND SEE IF <760	
6538	9F11	21 14 00	LXI H,LEN	
6539	9F14	19 . .	DAD D ;HL = X + LENGTH	
6540	9F15	11 CF 02	LXI D,719	
6541	9F18	CD 4A A3	CALL BNDCK1 ;HL = MIN(RTMOST,759)	
6542	9F1B	. . .	;NOW COMPUTE -LENGTH = -(RIGHT-LEFTMOST)	
6543	9F1B	CD 09 A3	CALL NEGATE ;HL = -RIGHTMOST	
6544	9F1E	EB . .	XCHG	
6545	9F1F	2A 69 90	LHLD XLEFT	
6546	9F22	EB . .	XCHG ;DE = LEFTMOST	
6547	9F23	19 . .	DAD D ;HL = -(RT - LEFT)	
6548	9F24	2B . .	DCX H	
6549	9F25	. . .	;SEND HORIZONTAL LENGTH TO HW	
6550	9F25	. . .	; FIRST WAIT FOR IDLE HARDWARE	
6551	9F25	CD 87 A2	CALL WAIT	
6552	9F28	22 1E 89	SHLD GC1DC	
6553	9F2B	. . .	;NOW SEND WA FOR LEFTMOST POINT	
6554	9F2B	. . .	;DE = XLEFT = XCOORD OF HORIZ VECTOR	
6555	9F2B	D5 . .	PUSH D ;SAVE X COORD	
6556	9F2C	2A 5F 90	LHLD GCY	
6557	9F2F	CD 5B 67	CALL MPY45 ;MULTIPLY Y BY 45	
6558	9F32	D1 . .	POP D ;RECALL X	
6559	9F33	1B . .	DCX D ;HW WANTS 1 LESS	
6560	9F34	CD 6F 67	CALL GETWA ;CONVERT TO WA	
6561	9F37	22 1A 89	SHLD GC1LO ;SEND BITS 0-11	
6562	9F3A	32 18 89	STA GC1HI ;SEND BITS 12-17	
6563	9F3D	. . .	;HORIZONTAL VECTOR FINISHED, NOW DO VERTICAL	
6564	9F3D	. . .	;COMPUTE STARTING POINT AND LENGTH	
6565	9F3D	. . .	;FIRST, COMPUTE BOTTOMMOST POINT	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 178
=====
6566     9F3D      2A 5F 90          LHL D GCY
6567     9F40      EB . .           XCHG                      ;DE = YCOORD
6568     9F41      21 EC FF         LXI H,MLEN                ;HL = LENGTH
6569     9F44      19 . .           DAD D                      ;HL = Y - LENGTH
6570     9F45      . . .           ;SEE IF BOTTOM MOST IS < 0
6571     9F45      7C . .           MOV A,H                   ;CHECK SIGN BIT
6572     9F46      B7 . .           ORA A
6573     9F47      F2 4D 9F        JP DGC020                 ;NO
6574     9F4A      21 00 00        LXI H,0                   ;YES--SET BOTTOMMOST = 0
6575     9F4D      . . .           DGC020 EQU $
6576     9F4D      22 67 90        SHLD YBOT                 ;STORE BOTTOMMOST POINT
6577     9F50      . . .           ;COMPUTE TOP POINT = Y + LENGTH
6578     9F50      21 14 00        LXI H,LEN                 ;HL = LENGTH
6579     9F53      19 . .           DAD D                      ;HL = Y + LENGTH
6580     9F54      . . .           ;SEE IF TOP COORDINATE IN BOUNDS (<360)
6581     9F54      11 67 01        LXI D,359
6582     9F57      CD 4A A3        CALL BNDCK1              ;HL = MIN(Y,359)
6583     9F5A      . . .           ;COMPUTE -LENGTH = -(TOP-BOTTOM)
6584     9F5A      CD 09 A3        CALL NEGATE              ;HL = -TOP
6585     9F5D      EB . .           XCHG
6586     9F5E      2A 67 90        LHL D YBOT
6587     9F61      EB . .           XCHG                      ;DE = BOTTOM
6588     9F62      19 . .           DAD D                      ;HL = -(TOP-BOTTOM)
6589     9F63      2B . .           DCX H
6590     9F64      . . .           ;SEND VERTICAL VECTOR LENGTH TO HW
6591     9F64      22 1C 89        SHLD GC2DC
6592     9F67      . . .           ;COMPUTE AND SEND WA FOR BOTTOM POINT
6593     9F67      . . .           ;DE = YBOT
6594     9F67      EB . .           XCHG                      ;HL = Y COORD OF BOTTOM
6595     9F68      2B . .           DCX H                      ;HW WANTS 1 LESS
6596     9F69      CD 5B 67        CALL MPY45               ;MULTIPLY BY 45
6597     9F6C      EB . .           XCHG
6598     9F6D      2A 61 90        LHL D GCX                ;X COORD OF BOTTOM
6599     9F70      EB . .           XCHG
6600     9F71      CD 6F 67        CALL GETWA               ;CONVERT TO WA
6601     9F74      . . .           ;SEND WA TO HARDWARE
6602     9F74      22 16 89        SHLD GC2LO              ;SEND 12 LSBITS
6603     9F77      32 14 89        STA GC2HI               ;SEND 6 MSBITS
6604     9F7A      . . .           ;SET MODE TO COMPLEMENT, TURN PATTERN OFF, AND
6605     9F7A      . . .           ;DRAW
6606     9F7A      3A B5 90        LDA CURMOD
6607     9F7D      E6 D8 .         ANI 330Q                 ;DELETE EJK, TURN SAMP OFF
6608     9F7F      . . .           ; SAMPLE BIT IS OFF TO INHIBIT PATTERN SHIFTS
6609     9F7F      F6 03 .         ORI 3Q                   ;SET MODE TO COMPLEMENT
6610     9F81      32 41 89        STA HCEJK               ;SEND TO HW
6611     9F84      21 B1 90        LXI H,GFLGS2            ;FETCH HW FLAGS
6612     9F87      7E . .           MOV A,M                  ;DONT SEND TO HW YET
6613     9F88      F6 11 .         ORI BUSY+DRWGC
6614     9F8A      . . .           ; F5 = 1 TO DRAW CURSOR
6615     9F8A      . . .           ; NOTE THAT FLAGS HAVE NOT BEEN SENT TO HW YET
=====

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 179
=====
6616     9F8A     77 . .      MOV  M,A      ;LET VR SEND FLAGS TO HW
6617     9F8B     C9 . .      RET
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 180
6619	9F8C	.	.	*****	
6620	9F8C	.	.	; GCTAB WAS MOVED TO ACCOMODATE CHECK SUMS	
6621	9F8C	.	.	*****	
6622	9F8C	.	.	GCTAB EQU \$	
6623	9F8C	00	00	DB 0,0,0,0	
6624	9F90	00	00	DB 0,0,1,0	
6625	9F94	01	00	DB 1,0,0,0	
6626	9F98	01	00	DB 1,0,1,0	
6627	9F9C	00	00	DB 0,0,377Q,377Q	
6628	9FA0	00	00	DB 0,0,0,0	
6629	9FA4	01	00	DB 1,0,377Q,377Q	
6630	9FA8	01	00	DB 1,0,0,0	
6631	9FAC	FF	FF	DB 377Q,377Q,0,0	
6632	9FB0	FF	FF	DB 377Q,377Q,1,0	
6633	9FB4	00	00	DB 0,0,0,0	
6634	9FB8	00	00	DB 0,0,1,0	
6635	9FBC	FF	FF	DB 377Q,377Q,377Q,377Q	
6636	9FC0	FF	FF	DB 377Q,377Q,0,0	
6637	9FC4	00	00	DB 0,0,377Q,377Q	
6638	9FC8	00	00	DB 0,0,0,0	
6639	9FCC	.	.	*****	
6640	9FCC	.	.	; GCXY--UPDATE X,Y OF CURSOR	
6641	9FCC	.	.	; ENTRY A='SPEED' = NO OF TIMES TO ADD INCREMENTS	
6642	9FCC	.	.	; EXIT ALL DESTROYED	
6643	9FCC	.	.	*****	
6644	9FCC	.	.	GCXY EQU \$	
6645	9FCC	F5	.	PUSH PSW ;SAVE SPEED	
6646	9FCD	.	.	;FETCH X AND Y INCREMENTS (+1,-1, OR 0) USING	
6647	9FCD	.	.	;ACTIVE CURSOR KEYS AS INDEX	
6648	9FCD	21	AF	LXI H,GFLGS4 ;FETCH ACTIVE KEYS	
6649	9FD0	5E	.	MOV E,M	
6650	9FD1	16	00	MVI D,0 ;DE = INDEX TO TABLE	
6651	9FD3	21	8C	LXI H,GCTAB ;BASE OF TABLE	
6652	9FD6	19	.	DAD D ;HL = POINTER TO INCREMENTS	
6653	9FD7	5E	.	MOV E,M ;LSBYTE X INC	
6654	9FD8	23	.	INX H	
6655	9FD9	56	.	MOV D,M ;DE = X INCREMENT	
6656	9FDA	23	.	INX H	
6657	9FDB	4E	.	MOV C,M ;LSBYTE Y INC	
6658	9FDC	23	.	INX H	
6659	9FDD	46	.	MOV B,M ;BC = Y INCREMENT	
6660	9FDE	.	.	;UPDATE COORDINATES BY ADDING INCREMENT	
6661	9FDE	.	.	;A IS LOOP COUNTER FOR NUMBER OF TIMES TO ADD	
6662	9FDE	.	.	;UPDATE XCOORD	
6663	9FDE	2A	CF	LHLD NEWGCX	
6664	9FE1	.	.	GXY010 EQU \$	
6665	9FE1	19	.	DAD D ;HL = X + INC	
6666	9FE2	3D	.	DCR A ;DECREMENT LOOP COUNTER	
6667	9FE3	F2	E1	JP GXY010 ;GO THRU AGAIN	
6668	9FE6	CD	60	CALL XCHECK ;INSURE IN BOUNDS	

13255

2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

				PAGE 181	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	
6669	9FE9	22	CF 90	SHLD NEWGCX	;STORE NEW X COORD
6670	9FEC	.	.	;UPDATE Y COORD	
6671	9FEC	2A	CD 90	LHLD NEWGCY	
6672	9FEF	F1	.	POP PSW	;RECALL SPEED
6673	9FF0	.	.	GXY020 EQU \$	
6674	9FF0	09	.	DAD B	;HL = Y + INC
6675	9FF1	3D	.	DCR A	;DCEREMENT LOOP COUNTER
6676	9FF2	F2	F0 9F	JP GXY020	;GO TRHU AGAIN
6677	9FF5	CD	69 A3	CALL YCHECK	;INSURE IN BOUNDS
6678	9FF8	.	.	;*****	
6679	9FF8	.	.	; ROM BREAK 5	
6680	9FF8	C3	02 A0	JMP ZBRK5C	
6681	9FF8	.	.	ORG ZBRK4+4000Q	
6682	A000	.	.	ZBRK5 EQU \$	
6683	A000	54	.	DB VERSN	
6684	A001	A0	.	DB ZBRK5/256	
6685	A002	.	.	ZBRK5C EQU \$	
6686	A002	.	.	;*****	
6687	A002	22	CD 90	SHLD NEWGCY	;STORE NEW Y COORD
6688	A005	.	.	;SET FLAGS TO INDICATE CURSOR HAS MOVED	
6689	A005	3E	61 .	MVI A,GCM1+GCM3+GCM4	
6690	A007	C3	40 A2	JMP STFLG5	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 182
6692	A00A	.	.	*****	
6693	A00A	.	.	; GCMON--UPDATE G-CURSOR COORDINATES	
6694	A00A	.	.	; CALLED FROM VERTICAL RETRACE ROUTINE	
6695	A00A	.	.	*****	
6696	A00A	.	.	GCMON EQU \$	
6697	A00A	3A	AF 90	LDA GFLGS4 ;ANY CURSOR KEYS ACTIVE?	
6698	A00D	B7	.	ORA A	
6699	A00E	C8	.	RZ ;NO--DONE	
6700	A00F	21	ED FB	LXI H,GCTIMR ;TIME TO UPDATE POSITION?	
6701	A012	35	.	DCR M ;UPDATE INITIAL REPEAT DELAY	
6702	A013	F0	.	RP ;NO	
6703	A014	36	00	MVI M,0 ;YES-STOP INITIAL DELAY TIME	
6704	A016	.	.	;UPDATE CURSOR SPEED	
6705	A016	.	.	;CURSOR STARTS OUT SLOW AND SPEEDS UP UNTIL NORMAL	
6706	A016	.	.	;RATE IS REACHED.	
6707	A016	.	.	;SPEED IS MERELY THE NUMBER OF DOTS THE CURSOR IS	
6708	A016	.	.	;MOVED.	
6709	A016	.	.	;LOAD MAX SPEED. IF IN ZOOM MODE, LOAD VALUE FROM	
6710	A016	.	.	;TABLE, OTHERWISE, USE NORMAL SPEED	
6711	A016	0E	18	MVI C,LOSPD ;ASSUME NOT IN ZOOM	
6712	A018	3A	AE 90	LDA GFLGS5 ;IN ZOOM MODE?	
6713	A01B	E6	02	ANI WANTZM	
6714	A01D	CA	24 A0	JZ GCM005 ;NO, LEAVE C = LOSPD	
6715	A020	21	E3 FB	LXI H,MAXSPD ;YES, LOAD VALUE FROM TABLE	
6716	A023	4E	.	MOV C,M	
6717	A024	.	.	GCM005 EQU \$	
6718	A024	21	EE FB	LXI H,SPEED ;CHECK CURRENT SPEED	
6719	A027	7E	.	MOV A,M ;UP TO NORMAL SPEED YET?	
6720	A028	B9	.	CMP C ;(COMPARE WITH MAX SPEED)	
6721	A029	D2	2D A0	JNC GCM010 ;YES--DONT INCREASE SPEED	
6722	A02C	34	.	INR M ;NO--CONTINUE TO SPEED UP	
6723	A02D	.	.	GCM010 EQU \$	
6724	A02D	.	.	; DIVIDE SPEED BY 8	
6725	A02D	0F	.	RRC	
6726	A02E	0F	.	RRC	
6727	A02F	0F	.	RRC	
6728	A030	E6	0F	ANI 170 ;DELETE ROTATED LSBITS	
6729	A032	.	.	;UPDATE X,Y OF CURSOR USING CURRENT SPEED	
6730	A032	C3	CC 9F	JMP GCXY	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 183
6732	A035	.	.	*****	
6733	A035	.	.	; GCKEYS--CHECK FOR GRAPHICS CURSOR KEY PRESSED	
6734	A035	.	.	; ENTRY A = KEY CODE FROM ASCII TABLE	
6735	A035	.	.	; EXIT CY => DONT PROCESS KEY ANY FURTHER	
6736	A035	.	.	; NO CY => CONTINUE TO PROCESS, NOT CURSOR	
6737	A035	.	.	; EXIT--ALL REGISTERS SAVED	
6738	A035	.	.	*****	
6739	A035	.	.	GCKEYS EQU \$	
6740	A035	.	.	;SEE IF CURSOR OR SPEED KEY	
6741	A035	FE	A1	CPI GCKEY ;RANGE FOR SPEED OR CURSOR?	
6742	A037	3F	.	CMC ;REVERSE SENSE OF CARRY	
6743	A038	D0	.	RNC ;NO, TOO SMALL. RET WITH NC	
6744	A039	FE	A6	CPI FSTKEY+1 ;>SPEED KEY?	
6745	A038	D0	.	RNC ;YES--RET WITH NO CARRY	
6746	A03C	.	.	;KEY IS EITHER SPEED OR CURSOR.DECIDE WHICH	
6747	A03C	F5	.	PUSH PSW ;SAVE EVERYTHING	
6748	A03D	C5	.	PUSH B	
6749	A03E	D5	.	PUSH D	
6750	A03F	E5	.	PUSH H	
6751	A040	.	.	;SEE IS FAST KEY	
6752	A040	FE	A5	CPI FSTKEY ;IS IT?	
6753	A042	CA	78 A0	JZ GCK020 ;YES--PROCESS	
6754	A045	3D	.	DCR A ;(ADJUST TO 0-3)	
6755	A046	E6	03	ANI 30 ;NO, ITS A CURSOR	
6756	A048	SF	.	MOV E,A ;SAVE CODE IN E	
6757	A049	16	00	MVI D,0 ;DE = INDEX TO FLAG TABLE	
6758	A04B	.	.	;SEE IF THIS IS THE FIRST ACTIVE KEY	
6759	A04B	3A	AF 90	LDA GFLGS4 ;FETCH ACTIVE KEYS	
6760	A04E	B7	.	ORA A ;Z FLAG SET IF THIS IS THE	
6761	A04F	F5	.	PUSH PSW ;FIRST. SAVE Z FLAGG	
6762	A050	.	.	;USING KEYCODE, SET PROPER BIT IN ACTIVE KEYS WORD	
6763	A050	F3	.	DI ;DONT ALLOW CURSOR UPDATES	
6764	A051	21	80 A0	LXI H,CURTAB ;BASE OF FLAG TABLE	
6765	A054	19	.	DAD D ;ADD KEYCODE = INDEX	
6766	A055	B6	.	ORA M ;MERGE BIT INTO ACTIVE KEYS	
6767	A056	32	AF 90	STA GFLGS4 ;SAVE NEW ACTIVE KEYS	
6768	A059	.	.	;IF KEY IS FIRST ONE PRESSED, SET INITIAL	
6769	A059	.	.	;REPEAT DELAY AND SPEED	
6770	A059	F1	.	POP PSW ;RECALL Z FLAG	
6771	A05A	C2	71 A0	JNZ GCK010 ;NOT THE FIRST ONE	
6772	A05D	.	.	;CURSOR KEY IS FIRST ONE PRESSED	
6773	A05D	21	ED FB	LXI H,GCTIMR ;SET INITAL DELAY	
6774	A060	36	10	MVI M,INTDLY	
6775	A062	AF	.	XRA A ;SET SPEED = 0	
6776	A063	CD	CC 9F	CALL GCXY ;DO INITIAL UPDATE	
6777	A066	21	EE FB	LXI H,SPEED ;IF FAST KEY NOT DOWN, SET	
6778	A069	3E	40	MVI A,HISPD ;INITIAL SLOW SPEED	
6779	A06B	BE	.	CMF M ;IN HIGH SPEED?	
6780	A06C	CA	71 A0	JZ GCK010 ;YES--DONT CHANGE	
6781	A06F	36	00	MVI M,0 ;NO--SET INITIAL SPEED	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 184
=====
6782     A071      . . .      GCK010 EQU $
6783     A071     FB . .      EI ;ALLOW CURSOR UPDATES
6784     A072     E1 . .      POP H ;RESTORE EVERYTHING
6785     A073     D1 . .      POP D
6786     A074     C1 . .      POP B
6787     A075     F1 . .      POP PSW
6788     A076     37 . .      STC ;RETURN WITH CY
6789     A077     C9 . .      RET
6790     A078      . . .      ;SPEED KEY PRESSED, CHANGE SPEED
6791     A078      . . .      GCK020 EQU $
6792     A078     21 EE FB    LXI H,SPEED
6793     A07B     36 40 .      MVI M,HISPD ;SET SPEED TO FAST
6794     A07D     C3 71 A0    JMP GCK010
6795     A080      . . .      CURTAB EQU $
6796     A080     04 08 10    DB 4Q,10Q,20Q,40Q
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 185
6798	A084	.	.	*****	
6799	A084	.	.	; RELGC--CHECK FOR CURSOR KEY RELEASED	
6800	A084	.	.	; CLEAR ACTIVE KEY FLAG OR UPDATE SPEED IF SO	
6801	A084	.	.	; ENTRY A = E = KEY NUMBER FROM GTKYMN	
6802	A084	.	.	; EXIT CY => DONT PROCESS KEY FURTHER	
6803	A084	.	.	; NC => CONTINUE TO PROCESS, NOT CURSOR	
6804	A084	.	.	; EXIT---ALL REGISTERS SAVED	
6805	A084	.	.	*****	
6806	A084	.	.	RELGC EQU \$	
6807	A084	D5	.	PUSH D ;SAVE REGISTERS	
6808	A085	E5	.	PUSH H	
6809	A086	F5	.	PUSH PSW	
6810	A087	.	.	; COMPUTE KEY CODE FROM LOWER CASE TABLE	
6811	A087	2A	2C 48	LHLD ZLWASC ;BASE OF LOWER CASE TABLE	
6812	A08A	16	00 .	MVI D,0 ;DE = INDEX TO PROPER KEY	
6813	A08C	19	.	DAD D ;HL = POINTER TO KEY CODE	
6814	A08D	7E	.	MOV A,M ;A = KEY CODE	
6815	A08E	.	.	; SEE IF CURSOR OR SPEED KEY	
6816	A08E	FE	A1 .	CPI GCKEY ;CURSOR KEY?	
6817	A090	DA	C1 A0	JC REL030 ;NO, TOO SMALL	
6818	A093	FE	A6 .	CPI FSTKEY+1 ;SPEED KEY?	
6819	A095	D2	C1 A0	JNC REL030 ;NO, TOO BIG	
6820	A098	.	.	;DECIDE WHETHER CURSOR OR SPEED	
6821	A098	FE	A5 .	CPI FSTKEY ;FAST KEY?	
6822	A09A	CA	B9 A0	JZ REL020 ;YES, PROCESS IT	
6823	A09D	.	.	;CURSOR KEY RELEASED, CLEAR ACTIVE FLAG	
6824	A09D	3D	.	DCR A ;(ADJUST TO 0-3)	
6825	A09E	E6	03 .	ANI 30	
6826	A0A0	5F	.	MOV E,A ;USE KEY CODE AS INDEX	
6827	A0A1	16	00 .	MVI D,0 ;TO CLEAR MASK TABLE	
6828	A0A3	21	C6 A0	LXI H,CLRTAB ;BASE OF TABLE	
6829	A0A6	19	.	DAD D ;HL = POINTER TO MASK	
6830	A0A7	3A	AF 90	LDA GFLGS4 ;ACTIVE KEY FLAGS	
6831	A0AA	A6	.	ANA M ;CLEAR PARTICULAR FLAG	
6832	A0AB	32	AF 90	STA GFLGS4	
6833	A0AE	.	.	REL010 EQU \$	
6834	A0AE	F1	.	POP PSW ;RESTORE ALL	
6835	A0AF	E1	.	POP H	
6836	A0B0	D1	.	POP D	
6837	A0B1	.	.	; TEST TO SEE IF THIS KEY IS REPEATING	
6838	A0B1	.	.	; IF SO, CLEAR THE REPEAT TIMER	
6839	A0B1	.	.	; HL = PTR TO REPEATING KEY, A = THIS KEY, D = 0	
6840	A0B1	BE	.	CMP M ;REPEATING?	
6841	A0B2	37	.	STC ;(CY => DONT PROCESS KEY)	
6842	A0B3	C0	.	RNZ ;NO, DONT CHANGE REPEAT TIME	
6843	A0B4	7A	.	MOV A,D ;YES, CLEAR TIMER	
6844	A0B5	32	EC 91	STA ZKBTMR	
6845	A0B8	C9	.	RET	
6846	A0B9	.	.	;SPEED KEY RELEASED, UPDATE SPEED	
6847	A0B9	.	.	REL020 EQU \$	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 186
=====
6848     A0B9      21  EE  FB          LXI  H,SPEED
6849     A0BC      36  00  .           MVI  M,0          ;SET TO SLOW SPEED
6850     A0BE      C3  AE  A0          JMP  REL010
6851     A0C1      .   .   .           ;KEY IS NEITHER CURSOR OR SPEED
6852     A0C1      .   .   .           REL030 EQU $
6853     A0C1      F1  .   .           POP  PSW          ;RESTORE ALL
6854     A0C2      E1  .   .           POP  H
6855     A0C3      D1  .   .           POP  D
6856     A0C4      B7  .   .           ORA  A           ;CLEAR CARRY => CONTINUE TO
6857     A0C5      C9  .   .           RET              ;PROCESS
6858     A0C6      .   .   .           CLRTAB EQU $
6859     A0C6      FB  F7  EF          DB   3730,3670,3570,3370
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 187
6861	A0CA	.	.	. ;*****	
6862	A0CA	.	.	. ; ZMUPDA -- ZOOM UPDATE, CALLED FROM VR ROUTINE	
6863	A0CA	.	.	. ;*****	
6864	A0CA	.	.	. ZMUPDA EQU \$	
6865	A0CA	.	.	. ; FIND OUT IF ANYTHING NEED TO BE UPDATED	
6866	A0CA	3A	E1	FB LDA MAG ;IS SIZE 1X ?	
6867	A0CD	B7	.	. ORA A	
6868	A0CE	C8	.	. RZ ;YES, NO UPDATE NECESSARY	
6869	A0CF	3A	AE	90 LDA GFLGSS ;HAS ZOOM BEEN TURNED ON?	
6870	A0D2	4F	.	. MOV C,A ;SAVE ZOOM STATUS	
6871	A0D3	E6	06	. ANI WANTZM+SUPRZM ;MASK STATUS BITS	
6872	A0D5	FE	02	. CPI WANTZM ;IS ZOOM ON, NOT SUPRESSED?	
6873	A0D7	C0	.	. RNZ ;NO, NO UPDATE NECESSARY	
6874	A0D8	.	.	. ;SEE IF THERE IS A NEW ZOOM SIZE. IF SO, USE THE	
6875	A0D8	.	.	. ;GRAPHICS CURSOR AS CENTER OF ZOOM AREA	
6876	A0D8	3E	80	. MVI A,NWZOOM ;CHECK NEW ZOOM FLAG	
6877	A0DA	A1	.	. ANA C ;IS IT ON?	
6878	A0DB	CA	F2	A0 JZ ZUD010 ;NO	
6879	A0DE	.	.	. ;NEW ZOOM SIZE, USE CURSOR AS ZOOM ADDRESS	
6880	A0DE	2F	.	. CMA ;CLEAR NEW ZOOM FLAG	
6881	A0DF	A1	.	. ANA C	
6882	A0E0	32	AE	90 STA GFLGSS	
6883	A0E3	2A	CF	90 LHLD NEWGCX ;SET ZA = CURSOR	
6884	A0E6	22	DF	FB SHLD ZX	
6885	A0E9	2A	CD	90 LHLD NEWGCY	
6886	A0EC	22	DD	FB SHLD ZY	
6887	A0EF	C3	43	A1 JMP ZUD040 ;GO SEND ZOOM PARAMETERS	
6888	A0F2	.	.	. ;NOT NEW ZOOM SIZE. IF CURSOR HAS CHANGED POSITION	
6889	A0F2	.	.	. ;MAKE SURE ITS STILL ON VISIBLE DISPLAY. IF NOT,	
6890	A0F2	.	.	. ;RECOMPUTE ZOOM ADDRESS	
6891	A0F2	.	.	. ZUD010 EQU \$	
6892	A0F2	3E	01	. MVI A,GCM1 ;DID CURSOR MOVE??	
6893	A0F4	A1	.	. ANA C	
6894	A0F5	C8	.	. RZ ;NO, NO UPDATE NEEDED	
6895	A0F6	2F	.	. CMA ;YES, CLEAR GCMOVD FLAG	
6896	A0F7	A1	.	. ANA C	
6897	A0F8	32	AE	90 STA GFLGSS	
6898	A0FB	.	.	. ;CHECK TO SE IF CURSOR IN DISPLAY BOUNDRIES	
6899	A0FB	.	.	. ;CASE 1 - IF ZX .LT. (GCX-360/MAG),LET ZX =	
6900	A0FB	.	.	. ;(GCX-360/M)	
6901	A0FB	2A	CF	90 LHLD NEWGCX	
6902	A0FE	EB	.	. XCHG ;DE = GCX	
6903	A0FF	.	.	. ;*****	
6904	A0FF	2A	E0	FA LHLD M360M2 ;HL = -360/MAG	
6905	A102	.	.	. ;*****	
6906	A102	19	.	. DAD D ;HL = GCX-360/M	
6907	A103	EB	.	. XCHG	
6908	A104	2A	DF	FB LHLD ZX ;HL = ZX, DE = GCX-360/M	
6909	A107	CD	55	A3 CALL BNDCK2 ;SEE IF ZX < GCX-360/M	
6910	A10A	FA	1C	A1 JM ZUD020 ;IF SO, DONT CHECK FURTHER	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 188
6911	A10D	.	.	. ;CASE 2 - IF ZX > GCX+ 360/M, LET ZX = GCX+360/M	
6912	A10D	EB	.	. XCHG ;DE = ZX	
6913	A10E	2A	CF	90 LHLD NEWGCX ;HL = GCX	
6914	A111	3A	EA	FB LDA P360M ;FETCH +360/MAG	
6915	A114	4F	.	. MOV C,A	
6916	A115	06	00	. MVI B,0 ;BC = +360/MAG	
6917	A117	09	.	. DAD B ;HL = GCX+360/M	
6918	A118	EB	.	. XCHG ;HL = ZX,DE = GCX+360/M	
6919	A119	CD	4A	A3 CALL BNDCK1 ;SEE IF HL > DE	
6920	A11C	.	.	. ZUD020 EQU \$	
6921	A11C	22	DF	FB SHLD ZX ;ZX OK IN RELATION TO GC NOW	
6922	A11F	.	.	. ;CHECK Y COORDINATE	
6923	A11F	.	.	. ;CASE 3 - IF ZY < (GCY-180/M), LET ZY = GCY-180/M	
6924	A11F	2A	CD	90 LHLD NEWGCY	
6925	A122	EB	.	. XCHG ;DE = GCY	
6926	A123	2A	E4	FB LHLD M180M ;HL = -180/MAG	
6927	A126	19	.	. DAD D	
6928	A127	EB	.	. XCHG ;DE = GCY-180/M,	
6929	A128	2A	DD	FB LHLD ZY ;HL = ZY	
6930	A12B	CD	55	A3 CALL BNDCK2 ;IS ZY < GCY-180/M?	
6931	A12E	FA	40	A1 JM ZUD030 ;IF SO, DONT CHECK FURTHER	
6932	A131	.	.	. ;CASE 4 - IF ZY > GCY+180/M, SET ZY = GCY+180/M	
6933	A131	EB	.	. XCHG ;DE = ZY	
6934	A132	2A	CD	90 LHLD NEWGCY ;HL = GCY	
6935	A135	.	.	. ;*****	
6936	A135	3A	DE	FA LDA P180M2 ;A = 180/MAG	
6937	A138	.	.	. ;*****	
6938	A138	4F	.	. MOV C,A	
6939	A139	06	00	. MVI B,0 ;BC = 180/MAG	
6940	A13B	09	.	. DAD B ;HL = GCY + 180/M	
6941	A13C	EB	.	. XCHG ;HL = ZY, DE = GCY+180/M	
6942	A13D	CD	4A	A3 CALL BNDCK1 ;SEE IF ZY > GCY+180/M	
6943	A140	.	.	. ZUD030 EQU \$	
6944	A140	22	DD	FB SHLD ZY ;ZY OK IN RELATION TO GC NOW	
6945	A143	.	.	. ;	
6946	A143	.	.	. ; HAVE CENTER OF ZOOM AREA ZX,ZY	
6947	A143	.	.	. ; CONVERT TO ZOOM ADDRESS, AND SEND WITH PARAMETER	
6948	A143	.	.	. ; S TO HW	
6949	A143	.	.	. ZUD040 EQU \$	
6950	A143	CD	72	A3 CALL ZCHECK ;INSURE ZX,ZY ARE VALID	
6951	A146	.	.	. ; COMPUTE WORD COUNT	
6952	A146	.	.	. ; WORD COUNT = XRIGHT-XLEFT + 1	
6953	A146	.	.	. ; COMPUTE XLEFT = (X-360/MAG) / 16	
6954	A146	2A	E6	FB LHLD M360M	
6955	A149	EB	.	. XCHG ;DE = -360/MAG	
6956	A14A	2A	DF	FB LHLD ZX	
6957	A14D	19	.	. DAD D ;HL = ZX - 360/MAG	
6958	A14E	22	5D	90 SHLD ZXTEMP ;SAVE ZX-360/MAG FOR HW ADDR	
6959	A151	0E	04	. MVI C,4 ;DIVIDE BY 16	
6960	A153	CD	1C	A3 CALL DIVHL ;HL = HL/16	


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 189
=====
6961     A156     22  69  90          SHLD XLEFT      ;HL=XLEFT=(X-360/M)/16
6962     A159     .   .   .          ;COMPUTE XRT = (X+360/MAG)/16
6963     A159     2A  EA  FB          LHLD P360M
6964     A15C     EB   .   .          XCHG            ;DE = +360/MAG
6965     A15D     2A  DF  FB          LHLD ZX
6966     A160     19   .   .          DAD D           ;HL = ZX + 360/MAG
6967     A161     0E  04  .          MVI C,4         ;DIVIDE BY 16
6968     A163     CD  1C  A3          CALL DIVHL      ;HL = HL/16
6969     A166     .   .   .          ;COMPUTE -WC = -(XRT-XLEFT+1)=XLEFT-XRT-1
6970     A166     CD  09  A3          CALL NEGATE     ;HL = -XRIGHT
6971     A169     EB   .   .          XCHG
6972     A16A     2A  69  90          LHLD XLEFT     ;HL=XLEFT, DE=-XRIGHT
6973     A16D     19   .   .          DAD D           ;HL = XLEFT-XRT
6974     A16E     2B   .   .          DCX H
6975     A16F     E5   .   .          PUSH H         ;SAVE WORD COUNT
6976     A170     .   .   .          ;COMPUTE AND SEND ZOOM ADDRESS
6977     A170     .   .   .          ;ZAX = ZX - 360/M = ZXTEMP
6978     A170     .   .   .          ;MUST COMPUTE ZAY = ZY + 180/M
6979     A170     2A  E8  FB          LHLD P180M
6980     A173     EB   .   .          XCHG            ;DE = +180/MAG
6981     A174     2A  DD  FB          LHLD ZY
6982     A177     19   .   .          DAD D           ;HL = ZY + 180/M
6983     A178     .   .   .          ;CONVERT TO SCREEN COORDINATES
6984     A178     CD  5B  67          CALL MPY45
6985     A17B     EB   .   .          XCHG
6986     A17C     2A  5D  90          LHLD ZXTEMP    ;ZOOM X COORD
6987     A17F     EB   .   .          XCHG            ;DE = ZAX, HL = ZAY
6988     A180     CD  6F  67          CALL GETWA
6989     A183     CD  87  A2          CALL WAIT       ;WAIT FOR IDLE HW
6990     A186     32  10  89          STA ZAH1       ;SEND ZOOM ADDRESS BITS 12-1
6991     A189     22  12  89          SHLD ZALO      ;SEND ZOOM ADDRESS BITS 0-11
6992     A18C     3E  0F  .          MVI A,170     ;COMPUTE PRESHIFT FROM 4 LSB
6993     A18E     A5   .   .          ANA L         ;OF ZOOM X
6994     A18F     2F   .   .          CMA           ;WANT ONES COMPLEMENT
6995     A190     32  05  89          STA PRESHF     ;SEND TO HW
6996     A193     E1   .   .          POP H         ;RECALL WORD COUNT
6997     A194     22  06  89          SHLD ZOOMWC    ;SEND TO HW
6998     A197     3A  E2  FB          LDA DCBYTE
6999     A19A     32  04  89          STA DCNTRL     ;SEND DISPLAY CONTROL BYTE
7000     A19D     .   .   .          ;COMPUTE REPEAT COUNT
7001     A19D     3A  E1  FB          LDA MAG
7002     A1A0     3D   .   .          DCR A         ;RC = MAGNIFICATION-1
7003     A1A1     2F   .   .          CMA           ;WANT CUMPLEMENT
7004     A1A2     6F   .   .          MOV L,A
7005     A1A3     26  FF  .          MVI H,377Q    ;SET MSBYTE -
7006     A1A5     22  08  89          SHLD ZOOMRC    ;SEND ZOOM REPEAT COUNT
7007     A1A8     .   .   .          ;CONTROLLER IS NOW LOADED WITH ZOOM PARAMETERS
7008     A1A8     .   .   .          ;SET FLAG TO INDICATE NEW ZOOM
7009     A1A8     21  B1  90          LXI H,GFLGS2  ;SET ZOOM FLAG
7010     A1AB     3E  0A  .          MVI A,ZOOM+NEWZM ;AND NEW ZOOM FLAG
=====

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 190
=====
7011     A1AD     B6 . .      ORA M
7012     A1AE     77 . .      MOV M,A ;STORE NEW HW FLAGS
7013     A1AF     . . .      ;LET VR ROUTINE SEND FLAGS TO HW
7014     A1AF     C9 . .      RET
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 191
7016	A1B0	.	.	*****	
7017	A1B0	.	.	;TINTR--TIMER INTERRUPT PROCESSING	
7018	A1B0	.	.	; ENTRY--DONT CARE	
7019	A1B0	.	.	; EXIT---ALL REGISTERS DESTROYED	
7020	A1B0	.	.	*****	
7021	A1B0	.	.	TINTR EQU \$	
7022	A1B0	.	.	;UPDATE THE SUPRESS TIMER, AND CLEAR SUPRESS BIT	
7023	A1B0	.	.	; IF TIME OUT	
7024	A1B0	21	EC FB	LXI H,SUPTMR	
7025	A1B3	35	.	DCR M ;UPDATE TIMER	
7026	A1B4	3E	08 .	MVI A,TIMSUP ;CLEAR SUPRESS BIT IF TIMEOU	
7027	A1B6	CC	39 A2	CZ CLFLG3	
7028	A1B9	.	.	; IF IN AUTO PLOT MODE, INSURE GOLD LED IS BLINKING	
7029	A1B9	CD	F4 B8	CALL CHEKAP ;IS AUTO PLOT ON?	
7030	A1BC	C8	.	RZ ;NO,DONE	
7031	A1BD	3A	F4 FF	LDA ZMDFL1 ;IN SELECT MODE?	
7032	A1C0	E6	20 .	ANI SELECT	
7033	A1C2	C0	.	RNZ ;YES, DONT BLINK GOLD LED	
7034	A1C3	21	0E FF	LXI H,ZBLFLG ;BLINK THE GOLD LED	
7035	A1C6	3E	20 .	MVI A,SELLED ;TO INDICATE AUTO PLOT IS ON	
7036	A1C8	B6	.	ORA M ;IS ON	
7037	A1C9	77	.	MOV M,A	
7038	A1CA	C9	.	RET	
7039	A1CB	.	.	*****	
7040	A1CB	.	.	; VR--SEE IF VERTCAL RETRACE HAS ARRIVED. IF SO,	
7041	A1CB	.	.	; DO CURSOR AND ZOOM UPDATES	
7042	A1CB	.	.	; ENTRY--DONT CARE	
7043	A1CB	.	.	; EXIT---ALL REGISTERS DESTROYED	
7044	A1CB	.	.	*****	
7045	A1CB	.	.	VR EQU \$	
7046	A1CB	3A	20 89	LDA HWSTAT ;CHECK VR FLAG	
7047	A1CE	E6	20 .	ANI VRFLAG ;FRAME DONE YET?	
7048	A1D0	C8	.	RZ ;NO	
7049	A1D1	32	61 89	STA VRESET ;YES--RESET VR FLAG	
7050	A1D4	CD	0A A0	CALL GCMON ;UPDATE CURSOR POSITION	
7051	A1D7	.	.	;COMPUTE CURSOR PARAMETERS IF NECESSARY, BUT	
7052	A1D7	.	.	;DONT INITIATE CURSOR DRAWING YET	
7053	A1D7	3A	B0 90	LDA GFLGS3 ;IS CURSOR ACTIVE?	
7054	A1DA	4F	.	MOV C,A ;SAVE STATUS	
7055	A1DB	E6	80 .	ANI WANTGC	
7056	A1DD	CA	14 A2	JZ VR030 ;NO--DO ZOOM UPDATE	
7057	A1E0	.	.	;DRAW THE CURSOR	
7058	A1E0	3E	40 .	MVI A,GCON ;IS CURSOR CURRENTLY ON?	
7059	A1E2	A1	.	ANA C	
7060	A1E3	C2	F9 A1	JNZ VR020 ;YES--TURN IT OFF	
7061	A1E6	.	.	; CURSOR IS TO BE TURNED ON	
7062	A1E6	.	.	; IF THE RB LINE IS WANTED, BUT IS NOT NOW ON,	
7063	A1E6	.	.	; TURN IT ON	
7064	A1E6	3E	20 .	MVI A,WANTRB ;RB LINE ACTIVE?	
7065	A1E8	A1	.	ANA C	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 192
7066	A1E9	CA	F3	A1	JZ VR010 ;NO, DO CURSOR UPDATE
7067	A1EC	CD	56	70	CALL RBON ;TURN RB LINE ON IF OFF
7068	A1EF	21	80	90	LXI H,GFLGS3 ;PU GFLGS3 BACK INTO C
7069	A1F2	4E	.	.	MOV C,M
7070	A1F3	.	.	.	VR010 EQU \$
7071	A1F3	CD	C2	9E	CALL GRCON1 ;TURN CURSOR ON
7072	A1F6	C3	14	A2	JMP VR030
7073	A1F9	.	.	.	VR020 EQU \$
7074	A1F9	.	.	.	; IF RBLINE IS ACTIVE, AND THE CURSOR HAS MOVED,
7075	A1F9	.	.	.	; TURN IT OFF
7076	A1F9	3E	20	.	MVI A,WANTRB ;IS IT ACTIVE?
7077	A1F8	A1	.	.	ANA C
7078	A1FC	CA	11	A2	JZ VR025 ;NO, JUST DO CURSOR
7079	A1FF	3A	AE	90	LDA GFLGS5 ;DID CURSOR MOVE?
7080	A202	E6	20	.	ANI GCM3
7081	A204	CA	11	A2	JZ VR025 ;NO, JUST DO CURSOR
7082	A207	CD	46	A2	CALL CLFLG5 ;CLEAR THE MOVED FLAG
7083	A20A	CD	7E	70	CALL RBOFF ;TURN RBLINE OFF
7084	A20D	21	80	90	LXI H,GFLGS3 ;PUT GFLGS3 BACK INTO C
7085	A210	4E	.	.	MOV C,M
7086	A211	.	.	.	VR025 EQU \$
7087	A211	CD	E8	9E	CALL GRCOF1 ;TURN THE CURSOR OFF
7088	A214	.	.	.	VR030 EQU \$
7089	A214	.	.	.	;DO ZOOM UPDATE
7090	A214	CD	CA	A0	CALL ZMUPDA
7091	A217	.	.	.	;SEND HWFLAGS TO HW. PROPER BITS HAVE BEEN SET
7092	A217	.	.	.	;TO DO CURSOR OR ZOOM IF REQUIRED
7093	A217	CD	74	A2	CALL SNDGCF ;SEND FLAGS TO DRAW
7094	A21A	.	.	.	;*****
7095	A21A	.	.	.	; RE-ENABLE RESETS
7096	A21A	.	.	.	;*****
7097	A21A	3E	02	.	MVI A,RSTON
7098	A21C	32	80	83	STA IOKBCO
7099	A21F	C9	.	.	RET
7100	A220	.	.	.	;*****
7101	A220	.	.	.	; NORST--DISALLOW RESETS
7102	A220	.	.	.	;*****
7103	A220	.	.	.	NORST EQU \$
7104	A220	3E	04	.	MVI A,RSTOFF
7105	A222	32	80	83	STA IOKBCO
7106	A225	C9	.	.	RET

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 193
=====
7108      A226      . . .      ;*****
7109      A226      . . .      ; UTILITY ROUTINES
7110      A226      . . .      ;*****
7111      A226      . . .      ;
7112      A226      . . .      ;
7113      A226      . . .      ;*****
7114      A226      . . .      ; STFLG1--SET BIT IN GFLGS1
7115      A226      . . .      ; ENTRY A = BITS TO BE SET
7116      A226      . . .      ; EXIT A = NEW GFLGS1
7117      A226      . . .      ; HL = ADDRESS OF GFLGS1
7118      A226      . . .      ;*****
7119      A226      . . .      STFLG1 EQU $
7120      A226      21 B2 90      LXI H,GFLGS1 ;ADDRESS OF FLAGS
7121      A229      B6 . .      ORA M ;MERGE IN BITS
7122      A22A      77 . .      MOV M,A ;STORE NEW FLAGS
7123      A22B      C9 . .      RET
7124      A22C      . . .      ;*****
7125      A22C      . . .      ; CLFLG1--CLEAR BITS IN GFLGS1
7126      A22C      . . .      ; ENTRY A = BITS TO BE CLEARED
7127      A22C      . . .      ; EXIT SAME AS STFLG1
7128      A22C      . . .      ;*****
7129      A22C      . . .      CLFLG1 EQU $
7130      A22C      21 B2 90      LXI H,GFLGS1
7131      A22F      2F . .      CMA
7132      A230      A6 . .      ANA M
7133      A231      77 . .      MOV M,A
7134      A232      C9 . .      RET
7135      A233      . . .      ;*****
7136      A233      . . .      ;STFLG3--SET BIT IN GFLGS3
7137      A233      . . .      ;*****
7138      A233      . . .      STFLG3 EQU $
7139      A233      21 B0 90      LXI H,GFLGS3
7140      A236      B6 . .      JRA M
7141      A237      77 . .      MOV M,A
7142      A238      C9 . .      RET
7143      A239      . . .      ;*****
7144      A239      . . .      ;CLFLG3--CLEAR FLAG IN GFLGS3
7145      A239      . . .      ;*****
7146      A239      . . .      CLFLG3 EQU $
7147      A239      21 B0 90      LXI H,GFLGS3
7148      A23C      2F . .      CMA
7149      A23D      A6 . .      ANA M
7150      A23E      77 . .      MOV M,A
7151      A23F      C9 . .      RET
7152      A240      . . .      ;*****
7153      A240      . . .      ; STFLG5 -- SET BITS IN GFLGS5
7154      A240      . . .      ;*****
7155      A240      . . .      STFLG5 EQU $
7156      A240      21 AE 90      LXI H,GFLGS5
7157      A243      B6 . .      ORA M
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 194
7158	A244	77	.	MOV M,A	
7159	A245	C9	.	RET	
7160	A246	.	.	*****	
7161	A246	.	.	; CLFLG5 -- CLEAR BITS IN GFLGS5	
7162	A246	.	.	*****	
7163	A246	.	.	CLFLG5 EQU \$	
7164	A246	21	AE 90	LXI H,GFLGS5	
7165	A249	2F	.	CMA	
7166	A24A	A6	.	ANA M	
7167	A24B	77	.	MOV M,A	
7168	A24C	C9	.	RET	
7169	A24D	.	.	*****	
7170	A24D	.	.	; STTKFL--SET BITS IN TEK FLAGS	
7171	A24D	.	.	*****	
7172	A24D	.	.	STTKFL EQU \$	
7173	A24D	21	AD 90	LXI H,TKFLGS	
7174	A250	B6	.	ORA M	
7175	A251	77	.	MOV M,A	
7176	A252	C9	.	RET	
7177	A253	.	.	*****	
7178	A253	.	.	; CLTKFL--CLEAR BITS IN TEK FLAGS	
7179	A253	.	.	*****	
7180	A253	.	.	CLTKFL EQU \$	
7181	A253	21	AD 90	LXI H,TKFLGS	
7182	A256	2F	.	CMA	
7183	A257	A6	.	ANA M	
7184	A258	77	.	MOV M,A	
7185	A259	C9	.	RET	
7186	A25A	.	.	*****	
7187	A25A	.	.	; STFLG6--SET FLAG IN GFLGS6	
7188	A25A	.	.	*****	
7189	A25A	.	.	STFLG6 EQU \$	
7190	A25A	21	97 90	LXI H,GFLGS6	
7191	A25D	B6	.	ORA M	
7192	A25E	77	.	MOV M,A	
7193	A25F	C9	.	RET	
7194	A260	.	.	*****	
7195	A260	.	.	; CLFLG6--CLEAR FLAG IN GFLGS6	
7196	A260	.	.	*****	
7197	A260	.	.	CLFLG6 EQU \$	
7198	A260	21	97 90	LXI H,GFLGS6	
7199	A263	2F	.	CMA	
7200	A264	A6	.	ANA M	
7201	A265	77	.	MOV M,A	
7202	A266	C9	.	RET	
7203	A267	.	.	*****	
7204	A267	.	.	; STFLG7--SET BITS IN GFLGS 7	
7205	A267	.	.	*****	
7206	A267	.	.	STFLG7 EQU \$	
7207	A267	21	96 90	LXI H,GFLGS7	

13255

13255/90010

2648A MICROCODE LISTING 'GR70'

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS          PAGE 195
=====
7208     A26A     B6 . .      ORA  M
7209     A26B     77 . .      MOV  M,A
7210     A26C     C9 . .      RET
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 196
=====
7212     A26D      . . .      ;*****
7213     A26D      . . .      ; CLFLG7--CLEAR BITS IN GFLGS7
7214     A26D      . . .      ;*****
7215     A26D      . . .      CLFLG7 EQU $
7216     A26D      21 96 90    LXI H,GFLGS7
7217     A270      2F . .     CMA
7218     A271      A6 . .     ANA M
7219     A272      77 . .     MOV M,A
7220     A273      C9 . .     RET
7221     A274      . . .      ;*****
7222     A274      . . .      ; SNDGCF--SEND GC AND ZOOM FLAGS TO HW, THEN
7223     A274      . . .      ; CLEAR THEM
7224     A274      . . .      ;*****
7225     A274      . . .      SNDGCF EQU $
7226     A274      21 81 90    LXI H,GFLGS2 ;FETCH HW FLAGS
7227     A277      4E . .     MOV C,M ;ARE ANY SET?
7228     A278      3E 19 .    MVI A,NEWZM+DRWGC+BUSY
7229     A27A      A1 . .     ANA C
7230     A27B      C8 . .     RZ
7231     A27C      CD 87 A2    CALL WAIT ;WAIT FOR IDLE HW
7232     A27F      79 . .     MOV A,C ;YES, SEND TO HW
7233     A280      32 20 89    STA HWFLGS ;SEND TO HW
7234     A283      E6 E6 .    ANI 377Q-DRWGC-BUSY-NEWZM ;CLEAR FLAGS
7235     A285      77 . .     MOV M,A ;SAVE HW FLAGS
7236     A286      C9 . .     RET
7237     A287      . . .      ;*****
7238     A287      . . .      ; WAIT --WAIT UNTIL CONTROLLER IS FREE
7239     A287      . . .      ;*****
7240     A287      . . .      WAIT EQU $
7241     A287      F5 . .     PUSH PSW
7242     A288      . . .      WAT010 EQU $
7243     A288      3A 20 89    LDA HWSTAT
7244     A28B      E6 01 .    ANI BUSY
7245     A28D      C2 88 A2    JNZ WAT010
7246     A290      F1 . .     POP PSW
7247     A291      C9 . .     RET
=====

```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 197
=====
7249     A292      .      .      .      ;*****
7250     A292      .      .      .      ; VRWAIT -- WAIT FOR VERTICAL RETRACE
7251     A292      .      .      .      ;*****
7252     A292      .      .      .      VRWAIT EQU $
7253     A292     F5      .      .      PUSH PSW          ;SAVE REGISTERS
7254     A293     E5      .      .      PUSH H
7255     A294     2E     03      .      MVI L,3          ;SET FOR 30 MS TIMEOUT
7256     A296     AF      .      .      XRA A            ;CLEAR INTERRUPT FLAG
7257     A297     32     F6     FF      STA INTFLG
7258     A29A     32     61     89      STA VRESET       ;CLEAR VR FLAG
7259     A29D      .      .      .      VRW010 EQU $
7260     A29D     3A     20     89      LDA HWSTAT       ;FETCH STATUS
7261     A2A0     E6     20      .      ANI VRFLAG       ;VR YET?
7262     A2A2     C2     B4     A2      JNZ VRW020       ;YES, EXIT
7263     A2A5     3A     F6     FF      LDA INTFLG       ;NO, CHECK INTERRUPT FLAG
7264     A2A8     D6     03      .      SUI TMRINT       ;TIMER INTERRUPT YET?
7265     A2AA     C2     9D     A2      JNZ VRW010       ;NO, CONTINUE LOOPING
7266     A2AD     32     F6     FF      STA INTFLG       ;YES, CLEAR INTERRUPT FLAG
7267     A2B0     2D      .      .      DCR L            ;TIME OUT YET?
7268     A2B1     C2     9D     A2      JNZ VRW010       ;NO
7269     A2B4      .      .      .      VRW020 EQU $
7270     A2B4     E1      .      .      POP H            ;RESTORE REGISTERS
7271     A2B5     F1      .      .      POP PSW
7272     A2B6     C9      .      .      RET
7273     A2B7      .      .      .      ;*****
7274     A2B7      .      .      .      ; ANCHK--CHECK TO SEE IF A/N DISPLAY OR CURSOR IS
7275     A2B7      .      .      .      ; BEING INHIBITED BEFORE STORE TO IOCRRW
7276     A2B7      .      .      .      ; MSGON BIT IN GFLGS1 INDICATES MESSAGE, SOFT
7277     A2B7      .      .      .      ; KEYS, OR AUTOPLLOT MENU IS UP, SO INHIBIT
7278     A2B7      .      .      .      ; SHOULD BE OVERRIDDEN
7279     A2B7      .      .      .      ; ENTRY--A = VALUE TO BE STORED TO IOCRRW
7280     A2B7      .      .      .      ; EXIT---ALL REGISTERS SAVED
7281     A2B7      .      .      .      ;*****
7282     A2B7      .      .      .      ANCHK EQU $
7283     A2B7     F5      .      .      PUSH PSW
7284     A2B8     C5      .      .      PUSH B
7285     A2B9     3A     B2     90      LDA GFLGS1       ;GET INHIBIT FLAGS
7286     A2BC     4F      .      .      MOV C,A          ;SAVE FLAGS
7287     A2BD     E6     10      .      ANI MSGON        ;IS MESSAGE BIT SET?
7288     A2BF     C2     CB     A2      JNZ ANC005       ;YES, DO THE STORE
7289     A2C2     3E     60      .      MVI A,AVINHB+ACINHB ;CURSOR OR DISPLAY
7290     A2C4     A1      .      .      ANA C            ;INHIBITED?
7291     A2C5     CA     CB     A2      JZ ANC005        ;NO, DO THE STORE
7292     A2C8     C1      .      .      POP B            ;RESTORE REGISTERS
7293     A2C9     F1      .      .      POP PSW
7294     A2CA     C9      .      .      RET
7295     A2CB      .      .      .      ANC005 EQU $
7296     A2CB     C1      .      .      POP B
7297     A2CC     F1      .      .      POP PSW          ;STORE THE DATA
7298     A2CD     32     20     87      STA ZIOCRRW
=====

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS      PAGE 198
=====
7299     A2D0     C9 . . .              RET
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 199
=====
7301      A2D1      . . .      ;*****
7302      A2D1      . . .      ; FORMAT--CONVERT PARAMETERS FROM PRMBUF INTO
7303      A2D1      . . .      ; INTERNAL FORMAT
7304      A2D1      . . .      ; PARAMETERS ARE CONVERTED FROM 2 FIVE BIT CHUNKS
7305      A2D1      . . .      ; TO ONE 10 BIT VALUE
7306      A2D1      . . .      ; ENTRY HL = POINTE TO MSBYTE
7307      A2D1      . . .      ; EXIT HL = PARAMETER
7308      A2D1      . . .      ; DE = POINTER TO NEXT PARAMETER (HL + 2)
7309      A2D1      . . .      ; A DESTROYED
7310      A2D1      . . .      ;*****
7311      A2D1      . . .      FORMAT EQU $
7312      A2D1      7E . . .      MOV A,M ;A = MSBYTE, BITS 5-9
7313      A2D2      23 . . .      INX H
7314      A2D3      5E . . .      MOV E,M ;E = LSBYTE, BITS 0-4
7315      A2D4      0F . . .      RRC ;A = 00098765
7316      A2D5      0F . . .      RRC
7317      A2D6      0F . . .      RRC ;A = 76500098
7318      A2D7      57 . . .      MOV D,A ;SAVE MSBITS
7319      A2D8      E6 E0 . . .      ANI 340Q ;A = 76500000
7320      A2DA      B3 . . .      ORA E ;A = 76543210
7321      A2DB      5F . . .      MOV E,A ;E = NEW LSBYTE
7322      A2DC      7A . . .      MOV A,D ;A = 76500098
7323      A2DD      E6 03 . . .      ANI 3Q ;A = 00000098
7324      A2DF      57 . . .      MOV D,A ;D = NEW MSBYTE
7325      A2E0      23 . . .      INX H ;POINTER TO NEXT PRM
7326      A2E1      EB . . .      XCHG ;DE = POINTER, HL = PARAM
7327      A2E2      C9 . . .      RET
7328      A2E3      . . .      ;*****
7329      A2E3      . . .      ; LNGFMT--SIMILAR TO FORMAT. PARAMETERS ARE
7330      A2E3      . . .      ; CONVERTED FROM 3 FIVE-BIT CHUNKS TO ONE
7331      A2E3      . . .      ; 16 BIT VALUE. BIT 16 (SIGN BIT) IS SET BY
7332      A2E3      . . .      ; BIT 15
7333      A2E3      . . .      ; ENTRY HL = POINTER TO MSBYTE OF PARAMETER
7334      A2E3      . . .      ; EXIT HL = 16 BIT PARAMETER
7335      A2E3      . . .      ;*****
7336      A2E3      . . .      LNGFMT EQU $
7337      A2E3      E5 . . .      PUSH H ;SAVE POINTER TO MSBYTE
7338      A2E4      23 . . .      INX H ;POINTER TO 10 LSBITS
7339      A2E5      CD D1 A2 . . .      CALL FORMAT ;CONVERT FIRST 10 BITS
7340      A2E8      EB . . .      XCHG ;DE = 10 LSB OF PARAM
7341      A2E9      E1 . . .      POP H ;POINTER TO 5 MSB
7342      A2EA      7E . . .      MOV A,M ;FETCH THEM
7343      A2EB      87 . . .      ADD A ;SHIFT LEFT 2 PLACES
7344      A2EC      87 . . .      ADD A
7345      A2ED      B2 . . .      ORA D ;MERGE WITH LOWER 10 BITS
7346      A2EE      FE 40 . . .      CPI 100Q ;IS BIT 15 SET?
7347      A2F0      DA F5 A2 . . .      JC LNF010 ;NO, LEAVE BIT 16 0
7348      A2F3      F6 80 . . .      ORI 200Q ;YES-SET SIGN BIT (B16)
7349      A2F5      . . .      LNF010 EQU $
7350      A2F5      57 . . .      MOV D,A ;RESTORE MSBYTE
=====

```

13255
2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/7

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 200
=====
7351     A2F6     EB . .      XCHG                                           ;HL = NEW PARAMETER
7352     A2F7     C9 . .      RET
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 201
7354	A2F8	.	.	*****	
7355	A2F8	.	.	; PRMSTR--STORE VECTOR PARAMETER IN PRMBUF	
7356	A2F8	.	.	; USING PRMDEX AS INDEX	
7357	A2F8	.	.	*****	
7358	A2F8	.	.	PRMSTR EQU \$	
7359	A2F8	3A	B6 90	LDA PRMDEX	
7360	A2FB	5F	.	MOV E,A	
7361	A2FC	16	00	MVI D,0 ;DE = INDEX	
7362	A2FE	21	B9 90	LXI H,PRMBUF ;BASE OF PARAMETER BUFFER	
7363	A301	19	.	DAD D ;HL = LOCATION OF EMPTY SLOT	
7364	A302	3A	88 FF	LDA ZCHAR ;FETCH PARAMETER	
7365	A305	E6	1F	ANI 370 ;WANT 5 LSB ONLY	
7366	A307	77	.	MOV M,A ;STORE IT	
7367	A308	C9	.	RET	
7368	A309	.	.	*****	
7369	A309	.	.	; NEGATE--TWOS COMPLEMENT OF HL	
7370	A309	.	.	; EXIT HL = -HL, A DESTROYED	
7371	A309	.	.	*****	
7372	A309	.	.	NEGATE EQU \$	
7373	A309	.	.	;COMPLEMENT EVERY BIT AND ADD 1	
7374	A309	7D	.	MOV A,L	
7375	A30A	2F	.	CMA	
7376	A30B	6F	.	MOV L,A ;COMPLEMENT LSBYTE	
7377	A30C	7C	.	MOV A,H	
7378	A30D	2F	.	CMA	
7379	A30E	67	.	MOV H,A ;COMPLEMENT MSBYTE	
7380	A30F	23	.	INX H	
7381	A310	C9	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 202
7383	A311	.	.	;*****	
7384	A311	.	.	; DIVHL--HL = HL / (2**N), SIGN BIT INTACT	
7385	A311	.	.	; ENTRY C = N (RIGHT SHIFT COUNT)	
7386	A311	.	.	; ENTRY TO DIVHL1 => DIVIDE BY 2	
7387	A311	.	.	; ENTRY TO DIVHLR => DIVIDE BY 2 AND ROUND	
7388	A311	.	.	;*****	
7389	A311	.	.	DIVHLR EQU \$	
7390	A311	D5	.	PUSH D ;SAVE D	
7391	A312	11	01 00	LXI D,1 ;ROUND OFF FACTOR	
7392	A315	0E	01 .	MVI C,1 ;DIVIDE BY 2	
7393	A317	C3	20 A3	JMP DHL000	
7394	A31A	.	.	DIVHL1 EQU \$	
7395	A31A	0E	01 .	MVI C,1	
7396	A31C	.	.	DIVHL EQU \$	
7397	A31C	D5	.	PUSH D ;SAVE D	
7398	A31D	11	00 00	LXI D,0 ;ROUND OFF FACTOR	
7399	A320	.	.	DHL000 EQU \$	
7400	A320	7C	.	MOV A,H ;CHECK SIGN	
7401	A321	B7	.	ORA A	
7402	A322	F5	.	PUSH PSW ;SAVE SIGN	
7403	A323	FC	09 A3	CM NEGATE ;GET ABS VAL IF -	
7404	A326	.	.	; DO NOT ROUND IF VALUE IS 1, OR CLIPPER WILL	
7405	A326	.	.	; NEVER CONVERGE. IF VALUE IS 1, SIMPLY RETURN	
7406	A326	.	.	; ZERO	
7407	A326	2B	.	DCX H ;TEST FOR HL = 1	
7408	A327	7C	.	MOV A,H ;IF HL IS 0 NOW, THEN	
7409	A328	B5	.	ORA L ;IT WAS 1 BEFORE	
7410	A329	CA	3B A3	JZ DHL020 ;EXIT WITH HL = 0	
7411	A32C	23	.	INX H ;RESTORE H TO WHAT IT WAS	
7412	A32D	19	.	DAD D ;ADD ROUND OFF FACTOR	
7413	A32E	.	.	DHL010 EQU \$	
7414	A32E	0D	.	DCR C ;ALL SHIFTS DONE?	
7415	A32F	FA	3B A3	JM DHL020 ;YES, EXIT	
7416	A332	7C	.	MOV A,H ;ROTATE MSBYTE RIGHT	
7417	A333	1F	.	RAR	
7418	A334	67	.	MOV H,A	
7419	A335	7D	.	MOV A,L ;SHIFT LSBYTE	
7420	A336	1F	.	RAR	
7421	A337	6F	.	MOV L,A	
7422	A338	C3	2E A3	JMP DHL010 ;DO NEXT SHIFT	
7423	A338	.	.	DHL020 EQU \$	
7424	A338	F1	.	POP PSW ;RECALL SIGN	
7425	A33C	D1	.	POP D ;RESTORE D	
7426	A33D	F0	.	RP	
7427	A33E	C3	09 A3	JMP NEGATE ;CONVERT BACK TO -	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 203
=====
7429     A341      . . .      ;*****
7430     A341      . . .      ; BNDCHK--BOUNDS CHECK
7431     A341      . . .      ; IF HL < 0, HL IS SET TO 0
7432     A341      . . .      ; IF HL > DE, SET HL = DE. (M FLAG ALSO SET)
7433     A341      . . .      ; BNDCK1--IF HL > DE, HL = DE (DONT TEST FOR -)
7434     A341      . . .      ; A DESTROYED
7435     A341      . . .      ;*****
7436     A341      . . .      BNDCHK EQU $
7437     A341      . . .      ; TEST FOR HL < 0
7438     A341      7C . . .      MOV A,H          ;CHECK SIGN
7439     A342      B7 . . .      ORA A
7440     A343      F2 4A A3      JP BNDCK1        ;NOT NEG, OK
7441     A346      21 00 00      LXI H,0          ;NEGATIVE,SET TO 0
7442     A349      C9 . . .      RET
7443     A34A      . . .      BNDCK1 EQU $
7444     A34A      E5 . . .      PUSH H           ;SAVE REGISTERS
7445     A34B      D5 . . .      PUSH D
7446     A34C      7B . . .      MOV A,E         ;COMPUTE DE-HL
7447     A34D      95 . . .      SUB L           ;SUBTRACT LSBYTES
7448     A34E      7A . . .      MOV A,D
7449     A34F      9C . . .      SBB H           ;SUBTRACT MSBYTES
7450     A350      D1 . . .      POP D           ;RESTORE REGISTERS
7451     A351      E1 . . .      POP H
7452     A352      F0 . . .      RP              ;HL =< DE, LEAVE AS IS
7453     A353      EB . . .      XCHG           ;HL > DE, SWAP
7454     A354      C9 . . .      RET
7455     A355      . . .      ;*****
7456     A355      . . .      ; BNDCK2 -- BOUNDS CHECK
7457     A355      . . .      ; IF HL < DE, SET HL = DE. (M FLAG ALSO SET)
7458     A355      . . .      ; A DESTROYED
7459     A355      . . .      ;*****
7460     A355      . . .      BNDCK2 EQU $
7461     A355      E5 . . .      PUSH H           ;SAVE REGISTERS
7462     A356      D5 . . .      PUSH D
7463     A357      7D . . .      MOV A,L         ;COMPUTE HL - DE
7464     A358      93 . . .      SUB E           ;SUBTRACT LSBYTES
7465     A359      7C . . .      MOV A,H
7466     A35A      9A . . .      SBB D           ;SUBTRACT MSBYTES
7467     A35B      D1 . . .      POP D           ;RESTORE REGISTERS
7468     A35C      E1 . . .      POP H
7469     A35D      F0 . . .      RP              ;HL >= DE, LEAVE AS IS
7470     A35E      EB . . .      XCHG           ;HL < DE, SWAP
7471     A35F      C9 . . .      RET
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 204
7473	A360	.	.	*****	
7474	A360	.	.	; XCHECK--CHECK X VALUE TO SEE IF IN BOUNDS	
7475	A360	.	.	; ENTRY HL = X COORD	
7476	A360	.	.	; EXIT HL = X COORD GUARENTEED TO BE IN BNDS	
7477	A360	.	.	*****	
7478	A360	.	.	XCHECK EQU \$	
7479	A360	D5	.	PUSH D ;SAVE D	
7480	A361	11	CF 02	LXI D,719 ;MAX X VALUE	
7481	A364	CD	41 A3	CALL BNDCHK	
7482	A367	D1	.	POP D ;RESTORE D	
7483	A368	C9	.	RET	
7484	A369	.	.	*****	
7485	A369	.	.	; YCHECK--CHECK Y VALUE TO SEE IF IN BOUNDS	
7486	A369	.	.	; SAME AS XCHECK	
7487	A369	.	.	*****	
7488	A369	.	.	YCHECK EQU \$	
7489	A369	D5	.	PUSH D	
7490	A36A	11	67 01	LXI D,359 ;MAX Y VALUE	
7491	A36D	CD	41 A3	CALL BNDCHK	
7492	A370	D1	.	POP D	
7493	A371	C9	.	RET	
7494	A372	.	.	*****	
7495	A372	.	.	; ZCHECK -- INSURE ZOOM COORDINATES IN BOUNDS	
7496	A372	.	.	*****	
7497	A372	.	.	ZCHECK EQU \$	
7498	A372	.	.	;LEFT BOUNDRY = 360/MAG. IF ZX < XLEFT,SET ZX=XLFT	
7499	A372	2A	EA FB	LHLD P360M	
7500	A375	EB	.	XCHG ;DE = +360/MAG	
7501	A376	2A	DF FB	LHLD ZX ;HL = ZX	
7502	A379	CD	55 A3	CALL BNDCK2 ;SEE IF HL < DE	
7503	A37C	FA	8B A3	JM ZCK010 ;IF SO, DONT CHECK ANY MORE	
7504	A37F	.	.	;RT BOUNDRY = 719-360/M. IF ZX > XRT, SET ZX=XRT	
7505	A37F	EB	.	XCHG ;HL = 360/M	
7506	A380	.	.	*****	
7507	A380	2A	E0 FA	LHLD M360M2 ;HL = -360/MAG	
7508	A383	.	.	*****	
7509	A383	01	CF 02	LXI B,719	
7510	A386	09	.	DAD B	
7511	A387	EB	.	XCHG ;DE = 719-360/MAG, HL = ZX	
7512	A388	CD	4A A3	CALL BNDCK1 ;SEE IF HL > DE	
7513	A38B	.	.	ZCK010 EQU \$	
7514	A38B	22	DF FB	SHLD ZX ;ZX IN BOUNDS FOR SURE NOW	
7515	A38E	.	.	;BOTTOM BOUNDRY = 180/MAG. IF ZY < YBOT, ZY=YBOT	
7516	A38E	.	.	*****	
7517	A38E	2A	DE FA	LHLD P180M2 ;HL = 180/MAG	
7518	A391	.	.	*****	
7519	A391	EB	.	XCHG	
7520	A392	2A	DD FB	LHLD ZY ;HL = ZY	
7521	A395	CD	55 A3	CALL BNDCK2 ;SEE IF HL < DE	
7522	A398	FA	A7 A3	JM ZCK020 ;IF SO, DONT CHECK FURTHER	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 205
7523	A39B	.	;TOP BOUNDRY = 359-180/M. IF ZY > YTOP,SET ZY=YTOP	
7524	A39B	EB	XCHG	
7525	A39C	2A E4 FB	LHLD M180M ;HL = -180/MAG	
7526	A39F	01 67 01	LXI B,359	
7527	A3A2	09 . .	DAD B	
7528	A3A3	EB . .	XCHG ;DE = 359-180/M,HL = ZY	
7529	A3A4	CD 4A A3	CALL BNDCK1 ;SEE IF HL > DE	
7530	A3A7	. . .	ZCK020 EQU \$	
7531	A3A7	22 DD FB	SHLD ZY ;ZY IN BOUNDS FOR SURE NOW	
7532	A3AA	C9 . .	RET	
7533	A3AB	. . .	;*****	
7534	A3AB	. . .	; GETPAT--FETCH PROPER PATTERN BYTE FROM 8X8	
7535	A3AB	. . .	; PATTERN, ROTATE TO ALIGN, AND SAVE AS CURPAT	
7536	A3AB	. . .	; USE 3 LSBITS OF X AND Y TO DO SELECTION	
7537	A3AB	. . .	;*****	
7538	A3AB	. . .	GETPAT EQU \$	
7539	A3AB	3A 69 90	LDA XSTART ;STARTING POINT OF VEC	
7540	A3AE	E6 07 .	ANI 7Q ;WANT 3 LSB ONLY	
7541	A3B0	4F . .	MOV C,A ;SAVE IN C	
7542	A3B1	3A 67 90	LDA YSTART	
7543	A3B4	E6 07 .	ANI 7Q	
7544	A3B6	5F . .	MOV E,A ;SAVE Y LSBITS IN E	
7545	A3B7	. . .	; IF DELTAX = 0, USE X TO SELECT BYTE, Y TO ROTATE	
7546	A3B7	. . .	; IF DELTAY = 0, USE Y TO SELECT BYTE, X TO ROTATE	
7547	A3B7	. . .	;IF NEITHER = 0, USE SOLID PATTERN AS ERROR	
7548	A3B7	. . .	;(CANT ALIGN PATTERN BYTES)	
7549	A3B7	2A D4 90	LHLD DELTAY ;DELTAY = 0?	
7550	A3BA	7C . .	MOV A,H	
7551	A3BB	B5 . .	ORA L	
7552	A3BC	21 F7 FB	LXI H,HAPAT ;USE HORIZONTAL PATTERN BYTE	
7553	A3BF	CA D6 A3	JZ GTPAT1 ;YES--LEAVE C,E REGS AS IS	
7554	A3C2	2A D6 90	LHLD DELTAX ;DELTAX = 0?	
7555	A3C5	7C . .	MOV A,H	
7556	A3C6	B5 . .	ORA L	
7557	A3C7	21 EF FB	LXI H,VAPAT ;USE VERTICAL PATTERN BYTES	
7558	A3CA	CA D3 A3	JZ GTP010 ;YES--SWAP C AND E REGS	
7559	A3CD	. . .	;ERROR--LINE NOT HORIZ OR VERT. CANT ALIGN PATTERN	
7560	A3CD	3E FF .	MVI A,377Q ;SET PATTERN TO ALL ON	
7561	A3CF	32 B4 90	STA CURPAT	
7562	A3D2	C9 . .	RET	
7563	A3D3	. . .	GTP010 EQU \$	
7564	A3D3	79 . .	MOV A,C ;SWAP C AND E REGS	
7565	A3D4	4B . .	MOV C,E	
7566	A3D5	5F . .	MOV E,A	
7567	A3D6	. . .	;ALTERNATE ENTRY FOR AREA FILL	
7568	A3D6	. . .	GTPAT1 EQU \$	
7569	A3D6	16 00 .	MVI D,0	
7570	A3D8	. . .	;USE DE AND HL TO SELECT PATTERN BYTE	
7571	A3D8	19 . .	DAD D ;HL = POINTER TO BYTE	
7572	A3D9	7E . .	MOV A,M ;FETCH BYTE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 206
=====
7573     A3DA      . . .      ;HAVE PATTERN BYTE, NOW ROTATE TO ALIGN
7574     A3DA      . . .      GTP020 EQU $
7575     A3DA      0D . .      DCR C          ;COMPLETELY ROTATED?
7576     A3DB      FA E2 A3    JM GTP030      ;YES--DONE
7577     A3DE      07 . .      RLC          ;NO--DO IT AGAIN
7578     A3DF      C3 DA A3    JMP GTP020
7579     A3E2      . . .      GTP030 EQU $
7580     A3E2      32 B4 90    STA CURPAT    ;STORE AS CURRENT PATTERN
7581     A3E5      C9 . .      RET
7582     A3E6      . . .      ;*****
7583     A3E6      . . .      ; MOVEGC--MOVE GRAPHICS CURSOR TO CURRENT
7584     A3E6      . . .      ; PEN POSITION
7585     A3E6      . . .      ;*****
7586     A3E6      . . .      MOVEGC EQU $
7587     A3E6      2A DE 90    LHLD XCURR   ;X COORD
7588     A3E9      CD 60 A3    CALL XCHECK  ;INSURE IN BOUNDS
7589     A3EC      22 CF 90    SHLD NEWGCX
7590     A3EF      2A DC 90    LHLD YCURR   ;Y COORD
7591     A3F2      CD 69 A3    CALL YCHECK  ;INSURE IN BOUNDS
7592     A3F5      22 CD 90    SHLD NEWGCV
7593     A3F8      3E 21 .     MVI A,GCM1+GCM3 ;SET CURSOR-HAS-MOVED
7594     A3FA      C3 40 A2    JMP STFLG5   ;FLAGS
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 207
7596	A3FD	.	.	*****	
7597	A3FD	.	.	; GTPAT2--GETPAT USING AREA FILL COORDINATES	
7598	A3FD	.	.	*****	
7599	A3FD	.	.	GTPAT2 EQU \$	
7600	A3FD	3A	69 90	LDA XLEFT ;USE X COORD TO DETERMINE	
7601	A400	E6	07 .	ANI 7Q ;ROTATION	
7602	A402	4F	. .	MOV C,A	
7603	A403	3A	67 90	LDA YBOT ;USE Y COORD TO SELECT	
7604	A406	E6	07 .	ANI 7Q ;ROW	
7605	A408	5F	. .	MOV E,A	
7606	A409	21	F7 FB	LXI H,HAPAT ;8 BY 8 PATTERN ADDRESS	
7607	A40C	C3	D6 A3	JMP GTPAT1 ;CONTINUE AS WITH GTPAT	
7608	A40F	.	.	*****	
7609	A40F	.	.	; VSETUP--LOAD CONSTANT VECTOR PARAMETERS FOR	
7610	A40F	.	.	; HORIZONTAL OR VERTICAL LINE	
7611	A40F	.	.	; EXIT HL,A = 0	
7612	A40F	.	.	*****	
7613	A40F	.	.	VSETUP EQU \$	
7614	A40F	AF	. .	XRA A	
7615	A410	32	09 89	STA SLFTST ;DISABLE SELFTEST	
7616	A413	32	07 89	STA CONST	
7617	A416	3D	. .	DCR A ;A = -1	
7618	A417	32	0B 89	STA SELWA ;USE NEW WA	
7619	A41A	32	01 89	STA DRWDOT ;DRAW FIRST DOT	
7620	A41D	32	11 89	STA MSBD ;SET D = - ANYTHING	
7621	A420	21	00 00	LXI H,0 ;SET D1 = 0	
7622	A423	22	1E 89	SHLD D1	
7623	A426	AF	. .	XRA A ;SET A = 0	
7624	A427	C9	. .	RET	
7625	A428	.	.	*****	
7626	A428	.	.	; GETKEY--GET A KEY AFTER MONITORING TAPES	
7627	A428	.	.	; EXIT Z=> KEY, A = KEYCODE	
7628	A428	.	.	*****	
7629	A428	.	.	GETKEY EQU \$	
7630	A428	21	2F 28	LXI H,ZCTMON ;VECTOR TO TAPE MONITOR	
7631	A42B	CD	A5 00	CALL ZIORGO ;EXECUTE IF ROM THERE	
7632	A42E	C3	05 48	JMP ZGETKY ;GET THE KEY	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 208
7634	A431	.	.	*****	
7635	A431	.	.	; VIDEO1-- PUT VIDEOS IN THE FOLLOWING STATE:	
7636	A431	.	.	; 1. GRAPHICS INHIBITED	
7637	A431	.	.	; 2. A/N VIDEO AND CURSOR ENABLED	
7638	A431	.	.	; CALLED BY SOFT KEYS ON, AUTO PLOT MENU ON, AND	
7639	A431	.	.	; DSPMSG (REPLACE SCREEN)	
7640	A431	.	.	; ENTRY--DONT CARE	
7641	A431	.	.	; EXIT---ALL REGISTERS DESTROYED	
7642	A431	.	.	*****	
7643	A431	.	.	VIDEO1 EQU \$	
7644	A431	CD	DF 9D	CALL LEDCHK ;SELF TEST IN PROGRESS?	
7645	A434	C2	3C A4	JNZ VD1010 ;NO	
7646	A437	.	.	; SELF TEST ERROR MESSAGE --JUST INHIBIT GRAFIX	
7647	A437	AF	.	XRA A ;TURN GRAPHICS VIDEO OFF	
7648	A438	32	41 89	STA HCEJK	
7649	A438	C9	.	RET	
7650	A43C	.	.	VD1010 EQU \$	
7651	A43C	3E	01 .	MVI A,SUPRO ;SUPRESS THE CURSOR	
7652	A43E	CD	8A 9E	CALL SUPRGC	
7653	A441	3A	85 90	LDA CURMOD ;FETCH MODE	
7654	A444	E6	EF .	ANI -1-GVENAB ;DELETE VIDEO BIT	
7655	A446	CD	32 72	CALL SNDMOD ;SEND TO HW	
7656	A449	3E	10 .	MVI A,MSGON ;SET MESSAGE ON BIT	
7657	A44B	C3	26 A2	JMP STFLG1 ;TO OVERRIDE A/N INHIBIT	
7658	A44E	.	.	; A/N CURSOR AND DISPLAY ARE TURNED ON BY ZRSTDP	
7659	A44E	.	.	;	
7660	A44E	.	.	*****	
7661	A44E	.	.	; VIDEO2--RESTORE GRAPHICS, A/N VIDEO TO ORIGINAL	
7662	A44E	.	.	; STATE. CALLED ONLY BY ZRSTDP	
7663	A44E	.	.	; VIDEO1 MAY OR MAY NOT HAVE BEEN CALLED	
7664	A44E	.	.	; IF NOT, THIS IS EFFECTIVELY A NOP, SINCE VIDEO	
7665	A44E	.	.	; STATE WILL NOT HAVE CHANGED	
7666	A44E	.	.	; EXIT NZ => A/N VIDEO INHIBITED	
7667	A44E	.	.	; ENTRY--DONT CARE	
7668	A44E	.	.	; EXIT---ALL REGISTERS DESTROYED	
7669	A44E	.	.	*****	
7670	A44E	.	.	VIDEO2 EQU \$	
7671	A44E	3A	85 90	LDA CURMOD ;RESTORE VIDEO BIT	
7672	A451	CD	32 72	CALL SNDMOD	
7673	A454	CD	AB 9E	CALL ENAB0 ;RE-ENABLE THE CURSOR	
7674	A457	3E	10 .	MVI A,MSGON ;CLEAR MESSAGE BIT	
7675	A459	CD	2C A2	CALL CLFLG1	
7676	A45C	4F	.	MOV C,A ;LEAVE GFLGS1 IN C	
7677	A45D	E6	40 .	ANI ACINHB ;WAS CURSOR INHIBITED?	
7678	A45F	CA	67 A4	JZ VID010 ;NO	
7679	A462	3E	18 .	MVI A,ZMXROW+1 ;YES, PUT IT OFF SCREEN	
7680	A464	32	20 87	STA ZIOCRW	
7681	A467	.	.	VID010 EQU \$	
7682	A467	3E	20 .	MVI A,AVINHB ;A/N VIDEO INHIBITED?	
7683	A469	A1	.	ANA C	

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 209
=====
7684     A46A     C8  .  .                RZ                                ;NO.
7685     A46B     CD 29 6E              CALL ANVOF1                       ;YES, TURN IT OFF
7686     A46E     F6  FF .              ORI 377Q                          ;SET NZ
7687     A470     C9  .  .                RET
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 210
7689	A471	.	.	*****	
7690	A471	.	.	; KBFUNC--PROCESS COMANDS FROM GRAPHICS KEYPAD	
7691	A471	.	.	; ENTRY A = KEYCODE	
7692	A471	.	.	; (RANGE OF CODES IS 207B TO 227B)	
7693	A471	.	.	; EXIT---ALL REGISTERS DESTROYED	
7694	A471	.	.	*****	
7695	A471	.	.	KBFUNC EQU \$	
7696	A471	.	.	; DISALLOW MENU KEY IF IN GIN MODE	
7697	A471	4F	.	MOV C,A ;SAVE KEYCODE	
7698	A472	FE	92	CPI MUKEY ;IS IT THE MENU KEY?	
7699	A474	C2	7F A4	JNZ KBF000 ;NO	
7700	A477	3A	AD 90	LDA TKFLGS ;YES, IN GIN MODE?	
7701	A47A	E6	10	ANI GINMOD	
7702	A47C	C2	8B A4	JNZ KBF005 ;YES, IGNORE KEY	
7703	A47F	.	.	KBF000 EQU \$	
7704	A47F	.	.	;IF AUTOPLLOT MENU IS ON, ONLY ALLOW TOGGLE MENU	
7705	A47F	.	.	;KEY TO WORK	
7706	A47F	CD	EE B8	CALL MUCHK ;IS MENU ON?	
7707	A482	CA	90 A4	JZ KBF010 ;NO, ALLOW ALL KEYS	
7708	A485	3E	92	MVI A,MUKEY ;YES--IS IT THE MENU KEY?	
7709	A487	B9	.	CMP C	
7710	A488	CA	90 A4	JZ KBF010 ;YES, PROCESS KEY	
7711	A48B	.	.	KBF005 EQU \$	
7712	A48B	3E	09	MVI A,STPRPT ;NO, STOP KEY REPEAT AND EXI	
7713	A48D	C3	08 48	JMP ZKBCTL	
7714	A490	.	.	KBF010 EQU \$	
7715	A490	CD	D9 9D	CALL DFCHK ;IN DISPLAY FUNCTIONS?	
7716	A493	C2	9C A4	JNZ KBF015 ;YES, ALLOW SOFT KEYS UP	
7717	A496	CD	C6 00	CALL ZCHKSF ;SOFT KEY MENU UP?	
7718	A499	C2	8B A4	JNZ KBF005 ;YES, DONT ALLOW KEY	
7719	A49C	.	.	KBF015 EQU \$	
7720	A49C	.	.	; STOP KEY FROM REPEATING IF NOT ZOOM KEY	
7721	A49C	79	.	MOV A,C ;RECALL KEY	
7722	A49D	FE	8B	CPI ZINKEY ;ZOOM IN KEY?	
7723	A49F	CA	AE A4	JZ KBF020 ;YES, ALLOW REPEAT	
7724	A4A2	FE	8C	CPI ZOUTKY ;ZOOM OUT KEY?	
7725	A4A4	CA	AE A4	JZ KBF020 ;YES, ALLOW REPEAT	
7726	A4A7	F5	.	PUSH PSW ;SAVE KEY CODE	
7727	A4A8	3E	09	MVI A,STPRPT ;STOP REPEAT	
7728	A4AA	CD	08 48	CALL ZKBCTL	
7729	A4AD	F1	.	POP PSW ;RECALL KEY CODE	
7730	A4AE	.	.	KBF020 EQU \$	
7731	A4AE	D6	87	SUI LWRFUN ;SUBTRACT BASE OF FUNCTION	
7732	A4B0	87	.	ADD A ;MPY BY 2 TO GET INDEX	
7733	A4B1	5F	.	MOV E,A	
7734	A4B2	16	00	MVI D,0 ;DE = INDEX TO FUNC TABLE	
7735	A4B4	21	D4 A4	LXI H,KYBDTB ;BASE OF KEYPAD FUNC TABLE	
7736	A4B7	.	.	*****	
7737	A4B7	.	.	; IF IN DISPLAY FUNCTIONS, OR STRAP A OUT (SEND	
7738	A4B7	.	.	; ALL FUNCTION CODES) SET UP FOR DISPLAY FUNCTS	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 211
7739	A4B7	. . .	;*****	
7740	A4B7	CD D9 9D	CALL DFCHK ;IN DISPLAY FUCNTIONS?	
7741	A4BA	C2 C5 A4	JNZ KBF030 ;YES	
7742	A4BD	3A FB FF	LDA KBJMP1 ;NO, STRAP A OUT?	
7743	A4C0	E6 01 .	ANI AJMPR ;=> SEND FUNCTION CODES	
7744	A4C2	CA CB A4	JZ KBF040 ;NO, EXECUTE KEY CODE	
7745	A4C5	. . .	; SET UP TO GENERATE CHARACTER STRING INSTEAD OF	
7746	A4C5	. . .	; EXECUTING	
7747	A4C5	. . .	KBF030 EQU \$	
7748	A4C5	CD 9D 6C	CALL LBLOFF ;TURN LABEL OFF (USES LBLBUF	
7749	A4C8	21 F6 A4	LXI H,DFTAB ;USE DISPLAY FUNCTIONS TABLE	
7750	A4CB	. . .	KBF040 EQU \$	
7751	A4CB	19 . .	DAD D ;POINTER TO FUNCTION	
7752	A4CC	5E . .	MOV E,M ;FETCH FUNCTION ADDRESS	
7753	A4CD	23 . .	INX H	
7754	A4CE	56 . .	MOV D,M	
7755	A4CF	EB . .	XCHG ;HL = POINTER TO FUNCTION	
7756	A4D0	CD 20 A2	CALL NORST ;DISALLOW RESETS	
7757	A4D3	E9 . .	PCHL ;JUMP TO ROUTINE	
7758	A4D4	. . .	;	
7759	A4D4	. . .	;	
7760	A4D4	. . .	KYBDTB EQU \$	
7761	A4D4	18 A5 .	DW STOPKY ;207--STOP AUTOPLLOT, G TEXT	
7762	A4D6	1F A5 .	DW TGLGC ;210--TOGGLE GRAPHICS CURSOR	
7763	A4D8	73 A5 .	DW TGLRB ;211--TOGGLE RB LINE	
7764	A4DA	7E A5 .	DW TGLZM ;212--TOGGLE ZOOM MODE	
7765	A4DC	90 A5 .	DW ZOOMIN ;213--INCREASE ZOOM SIZE	
7766	A4DE	AA A5 .	DW ZMOUT ;214--DECREASE ZOOM SIZE	
7767	A4E0	03 A6 .	DW CLRKY ;215--CLEAR SCREEN,TEK 'PAGE	
7768	A4E2	B8 A5 .	DW TGLGVD ;216--TOGGLE GRAPHICS VIDEO	
7769	A4E4	C3 A5 .	DW TGLAN ;217--TOGGLE A/N VIDEO	
7770	A4E6	CE A5 .	DW KBDRAW ;220--DRAW TO CURSOR	
7771	A4E8	EA A5 .	DW KBMOVE ;221--MOVE TO CURSOR	
7772	A4EA	FA A5 .	DW TGLMU ;222--TOGGLE AUTOPLLOT MENU	
7773	A4EC	99 B7 .	DW APLTON ;223--START AUTOPLLOT	
7774	A4EE	EF B1 .	DW APAXES ;224--DRAW AUTOPLLOT AXES	
7775	A4F0	0C A6 .	DW STX ;225--START GRAPHICS TEXT	
7776	A4F2	86 A6 .	DW TXTANG ;226--SET TEXT ANGLE	
7777	A4F4	1B A6 .	DW TXTSIZ ;227--SET TEXT SIZE	
7778	A4F6	. . .	;*****	
7779	A4F6	. . .	; DISPLAY FUNCTIONS TABLE	
7780	A4F6	. . .	;*****	
7781	A4F6	. . .	DFTAB EQU \$	
7782	A4F6	93 A7 .	DW DFSTOP ;207--STOP KEY	
7783	A4F8	A0 A7 .	DW DFGC ;210--TOGGLE CURSOR	
7784	A4FA	B7 A7 .	DW DFRB ;211--TOGGLE RB LINE	
7785	A4FC	CE A7 .	DW DFZM ;212--TOGGLE ZOOM	
7786	A4FE	E5 A7 .	DW DFZIN ;213--ZOOM IN	
7787	A500	F2 A7 .	DW DFZOUT ;214--ZOOM OUT	
7788	A502	16 A8 .	DW DFCLR ;215--CLEAR	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 212
=====
```

7789	A504	1F A8 .	DW DFGVD ;216--TOGGLE G VIDEO
7790	A506	36 A8 .	DW DFAVD ;217--TOGGLE A/N VIDEO
7791	A508	0B A9 .	DW DFDRAW ;220--DRAW
7792	A50A	C8 A8 .	DW DFMOV ;221--MOVE
7793	A50C	4D A8 .	DW DFMENU ;222--APMENU
7794	A50E	56 A8 .	DW DFAPON ;223--AUTO PLOT ON
7795	A510	5F A8 .	DW DFAXES ;224--AXES
7796	A512	68 A8 .	DW DFSTX ;225--TEXT
7797	A514	83 A8 .	DW DFTANG ;226--T ANG
7798	A516	B0 A8 .	DW DFTSIZ ;227--T SIZE


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 213
=====
7800      A518      . . .      ;*****
7801      A518      . . .      ; GRAPHICS FUNCTIONS INVOKED FROM KEYPAD
7802      A518      . . .      ;*****
7803      A518      . . .      ;
7804      A518      . . .      ;
7805      A518      . . .      ;*****
7806      A518      . . .      ; STOP--TURN MENU OFF, STOP GRAPHICS TEXT
7807      A518      . . .      ; TURN AUTO PLOT OFF
7808      A518      . . .      ;*****
7809      A518      . . .      STOPKY EQU $
7810      A518      CD 82 88      CALL APLTOF      ;AUTO PLOT OFF
7811      A518      CD 1E 76      CALL GTXOF1      ;GRAPHICS TEXT OFF
7812      A51E      C9 . .      RET
7813      A51F      . . .      ;*****
7814      A51F      . . .      ; TGLGC--TOGGLE GRAPHICS CURSOR
7815      A51F      . . .      ;*****
7816      A51F      . . .      TGLGC EQU $
7817      A51F      . . .      ; IF CONTROL KEY DOWN, DISPLAY CURSOR COORDS
7818      A51F      3A 12 FF      LDA ZCTCOL      ;CHECK CONTROL COLUMN
7819      A522      E6 01 .      ANI ZCTLKY      ;CONRTROL KEY DOWN?
7820      A524      C2 32 A5      JNZ LOCATE      ;YES, DISPLAY CURSOR LOCATIO
7821      A527      3A B0 90      LDA GFLGS3      ;IS CURSOR ON NOW?
7822      A52A      E6 80 .      ANI WANTGC
7823      A52C      C2 02 70      JNZ TGCOF1      ;YES, TURN IT OFF
7824      A52F      C3 D6 6F      JMP TGCON1      ;NO, TURN IT ON
7825      A532      . . .      ;*****
7826      A532      . . .      ; LOCATE--DISPLAY CURSOR POSITON UNTIL RETURN
7827      A532      . . .      ; KEY OR CURSOR KEY HIT
7828      A532      . . .      ;*****
7829      A532      . . .      LOCATE EQU $
7830      A532      CD 9D 6C      CALL LBLOFF      ;CLEAR PENDING LABEL
7831      A535      21 5F A5      LXI H,GCMMSG      ;POINTER TO CURSOR MESSAGE
7832      A538      22 F1 FF      SHLD ZMSGP1
7833      A53B      21 3F FB      LXI H,NUMBUF+50 ;BUFFER FOR CURSOR POSITON
7834      A53E      22 EF FF      SHLD ZMSGP2
7835      A541      CD F9 A8      CALL DFM1      ;GET CURSOR POSITION
7836      A544      36 CE .      MVI M,ZEOP      ;STORE END OF MESSAGE
7837      A546      CD ED 00      CALL ZDPMG2      ;PUT UP CURSOR, ALLOW G VIDE
7838      A549      . . .      ; WAIT FOR RETURN OR CURSOR KEY
7839      A549      CD 28 A4      CALL GETKEY      ;GET A KEY
7840      A54C      C2 59 A5      JNZ LCT010      ;NONE HIT
7841      A54F      FE EF .      CPI SFTCR      ;RETURN?
7842      A551      CA 43 00      JZ ZRSTDP      ;YES, RESTORE DISPLAY
7843      A554      FE 88 .      CPI CURKEY      ;CURSOR KEY?
7844      A556      CA 43 00      JZ ZRSTDP      ;YES, RESTORE DISPLAY
7845      A559      . . .      LCT010 EQU $
7846      A559      CD CB A1      CALL VR      ;DO CURSOR UPDATES
7847      A55C      C3 32 A5      JMP LOCATE
7848      A55F      . . .      GCMMSG EQU $
7849      A55F      82 47 52      DB IVON,'GRAPHICS CURSOR ',0
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 214
7851	A573	.	.	.	;*****	
7852	A573	.	.	.	; TGLRB--TOGGLE RUBER BAND LINE	
7853	A573	.	.	.	;*****	
7854	A573	.	.	.	TGLRB EQU \$	
7855	A573	3A	B0	90	LDA GFLGS3 ;IS RB LINE ON?	
7856	A576	E6	20	.	ANI WANTRB	
7857	A578	C2	4E	70	JNZ TRBOF1 ;YES, TURN IT OFF	
7858	A578	C3	37	70	JMP TRBON1 ;NO, TURN IT ON	
7859	A57E	.	.	.	;*****	
7860	A57E	.	.	.	; TGLZM--TOGGLE ZOOM MODE	
7861	A57E	.	.	.	;*****	
7862	A57E	.	.	.	TGLZM EQU \$	
7863	A57E	3A	E1	FB	LDA MAG ;TURN ZOOM OFF IF AT 1:1	
7864	A581	B7	.	.	ORA A ;AT 1:1 ??	
7865	A582	CA	5E	6E	JZ ZOFF1 ;YES, TURN IT OFF	
7866	A585	3A	AE	90	LDA GFLGS5 ;IN ZOOM MODE NOW?	
7867	A588	E6	02	.	ANI WANTZM	
7868	A58A	C2	5E	6E	JNZ ZOFF1 ;YES, TURN IT OFF	
7869	A58D	C3	41	6E	JMP ZON1 ;NO, TURN IT ON	
7870	A590	.	.	.	;*****	
7871	A590	.	.	.	; ZOOMIN--INCREASE ZOOM SIZE	
7872	A590	.	.	.	;*****	
7873	A590	.	.	.	ZOOMIN EQU \$	
7874	A590	.	.	.	; IF NOT IN ZOOM MODE, TURN ZOOM ON AT 2X	
7875	A590	3A	AE	90	LDA GFLGS5 ;IN ZOOM MODE?	
7876	A593	E6	02	.	ANI WANTZM	
7877	A595	C2	A0	A5	JNZ ZIN010 ;YES, UPDATE SIZE	
7878	A598	3E	01	.	MVI A,1 ;NO, SET SIZE TO 2X	
7879	A59A	CD	9E	6E	CALL NWSIZE	
7880	A59D	C3	41	6E	JMP ZON1 ;TURN ZOOM ON	
7881	A5A0	.	.	.	ZIN010 EQU \$	
7882	A5A0	3A	E1	FB	LDA MAG ;FETCH CURRENT SIZE	
7883	A5A3	3C	.	.	INR A ;TRY TO GO BIGGER	
7884	A5A4	FE	10	.	CPI MAXMAG ;TOO BIG?	
7885	A5A6	D0	.	.	RNC ;YES, LEAVE ZOOM AS IS	
7886	A5A7	C3	9E	6E	JMP NWSIZE ;NO, UPDATE SIZE	
7887	A5AA	.	.	.	;*****	
7888	A5AA	.	.	.	; ZMOUT--DECREASE ZOOM SIZE	
7889	A5AA	.	.	.	;*****	
7890	A5AA	.	.	.	ZMOUT EQU \$	
7891	A5AA	3A	AE	90	LDA GFLGS5 ;IF NOT IN ZOOM MODE	
7892	A5AD	E6	02	.	ANI WANTZM ;DONT CHANGE SIZE	
7893	A5AF	C8	.	.	RZ	
7894	A5B0	3A	E1	FB	LDA MAG ;FETCH CURRENT SIZE	
7895	A5B3	3D	.	.	DCR A ;TRY TO GO SMALLER	
7896	A5B4	F8	.	.	RM ;TOO SMALL, LEAVE AS IS	
7897	A5B5	C3	9E	6E	JMP NWSIZE ;UPDATE SIZE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 215
=====
7899      A5B8      . . .      ;*****
7900      A5B8      . . .      ; TGLGVD--TOGGLE GRAPHICS VIDEO
7901      A5B8      . . .      ;*****
7902      A5B8      . . .      TGLGVD EQU $
7903      A5B8      3A B5 90      LDA CURMOD      ;IS VIDEO ON NOW?
7904      A5B8      E6 10 .      ANI GVENAB
7905      A5B8      C2 ED 6D      JNZ GVOFF1      ;YES, TURN IT OFF
7906      A5C0      C3 AC 6D      JMP GVON1       ;NO, TURN IT ON
7907      A5C3      . . .      ;*****
7908      A5C3      . . .      ; TGLAN--TOGGLE A/N VIDEO
7909      A5C3      . . .      ;*****
7910      A5C3      . . .      TGLAN EQU $
7911      A5C3      3A B2 90      LDA GFLGS1      ;IS VIDEO ON NOW?
7912      A5C6      E6 20 .      ANI AVINHB      ;CHECK INHIBIT BIT
7913      A5C8      C2 18 6E      JNZ ANVON1      ;NO, TURN IT ON
7914      A5C8      C3 29 6E      JMP ANVOF1      ;YES, TURN IT OFF
7915      A5CE      . . .      ;*****
7916      A5CE      . . .      ; KBDRAW--DRAW VECTOR FROM CURRENT POINT TO CURSOR
7917      A5CE      . . .      ;*****
7918      A5CE      . . .      KBDRAW EQU $
7919      A5CE      CD 81 A9      CALL GCCHK      ;IS CURSOR OFF?
7920      A5D1      C0 . . .      RNZ             ;YES--DONT DRAW
7921      A5D2      CD F4 B8      CALL CHEKAP     ;IS AUTO PLOT ON?
7922      A5D5      C0 . . .      RNZ             ;YES, DONT DRAW
7923      A5D6      3E 01 .      MVI A,MOVE     ;CLEAR MOVE FLAG
7924      A5D8      CD 2C A2      CALL CLFLG1
7925      A5DB      . . .      KBD010 EQU $
7926      A5DB      2A CF 90      LHLD NEWGCX     ;SET NEW POINT TO CURSOR
7927      A5DE      22 DA 90      SHLD XNEW      ;LOCATION
7928      A5E1      2A CD 90      LHLD NEWGCY
7929      A5E4      22 D8 90      SHLD YNEW
7930      A5E7      C3 DB 65      JMP VECTOR      ;DRAW THE VECTOR
7931      A5EA      . . .      ;*****
7932      A5EA      . . .      ; KBMOVE--MOVE VECTOR ENDPOINT TO CURSOR
7933      A5EA      . . .      ;*****
7934      A5EA      . . .      KBMOVE EQU $
7935      A5EA      CD 81 A9      CALL GCCHK      ;IS CURSOR OFF?
7936      A5ED      C0 . . .      RNZ             ;YES--DONT MOVE
7937      A5EE      CD F4 B8      CALL CHEKAP     ;IS AUTO PLOT ON?
7938      A5F1      C0 . . .      RNZ             ;YES, DONT MOVE
7939      A5F2      3E 01 .      MVI A,MOVE     ;SET THE MOVE FLAG
7940      A5F4      CD 26 A2      CALL STFLG1
7941      A5F7      C3 DB A5      JMP KBD010      ;DO THE MOVE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 216
7943	A5FA	.	.	.	;*****
7944	A5FA	.	.	.	; TGLMU--TOGGLE AUTO PLOT MENU
7945	A5FA	.	.	.	;*****
7946	A5FA	.	.	.	TGLMU EQU \$
7947	A5FA	CD	EE	88	CALL MUCHK ;MENU ON NOW?
7948	A5FD	C2	D9	AA	JNZ APMUOF ;YES, TURN IT OFF
7949	A600	C3	C9	AA	JMP APMUON ;NO, TURN IT ON
7950	A603	.	.	.	;*****
7951	A603	.	.	.	; CLRKY--CLEAR SCREEN IF IN NORMAL MODE, ALSO
7952	A603	.	.	.	; EXIT GRAPHICS IF IN TEK MODE
7953	A603	.	.	.	;*****
7954	A603	.	.	.	CLRKY EQU \$
7955	A603	CD	56	69	CALL CHKTEK ;IN TEK MODE?
7956	A606	C2	1D	6A	JNZ PAGE ;YES, DO A 'PAGE'
7957	A609	C3	78	6D	JMP GCLR1 ;NO, JUST CLEAR SCREEN
7958	A60C	.	.	.	;*****
7959	A60C	.	.	.	; STX--START GRAPHICS TEXT FROM KEYBOARD
7960	A60C	.	.	.	;*****
7961	A60C	.	.	.	STX EQU \$
7962	A60C	CD	F7	75	CALL GTXON1 ;TURN GTEXT ON
7963	A60F	CD	D6	6F	CALL TGCON1 ;TURN THE CURSOR ON
7964	A612	CD	5C	69	CALL CKSCLO ;DONT CHANGE IF SCALED TEK O
7965	A615	C0	.	.	RNZ
7966	A616	3E	04	.	MVI A,4 ;SET JAM PATTERN MODE
7967	A618	C3	20	72	JMP SETMD1

=====					PAGE 217	
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	
=====						
7969	A61B	.	.	.	;*****	
7970	A61B	.	.	.	; TXTSIZ--CHANGE TEXT SIZE FROM KEYBOARD	
7971	A61B	.	.	.	;*****	
7972	A61B	.	.	.	TXTSIZ EQU \$	
7973	A61B	CD	5C	69	CALL CKSCLO	;DONT CHANGE IF SCALED TEK 0
7974	A61E	C2	5B	A7	JNZ TXT1	;IGNORE IF SCALED TEK MODE
7975	A621	AF	.	.	XRA A	;CLEAR ANY PENDING LABEL
7976	A622	32	74	90	STA LBLCTR	
7977	A625	.	.	.	; PUT UP MESSAGE	
7978	A625	.	.	.	TXS005 EQU \$	
7979	A625	21	72	A6	LXI H,SIZMSG	;FIXED PART OF MESSAGE
7980	A628	22	F1	FF	SHLD ZMSGP1	
7981	A62B	21	0D	FB	LXI H,NUMBUF	;WHERE CURRENT SIZE WILL GO
7982	A62E	22	EF	FF	SHLD ZMSGP2	
7983	A631	.	.	.	; LOAD CURRENT SIZE	
7984	A631	3A	DA	FB	LDA TXMAG	;CURRENT TEXT SIZE
7985	A634	C6	31	.	ADI 61Q	;CONVERT TO ASCII
7986	A636	77	.	.	MOV M,A	;STORE IN BUFFER
7987	A637	23	.	.	INX H	
7988	A638	36	20	.	MVI M,40Q	;STORE TERMINATING SPACE
7989	A63A	23	.	.	INX H	
7990	A63B	36	CE	.	MVI M,ZEOP	;STORE THE EOP
7991	A63D	37	.	.	STC	;REPLACE DISPLAY WITH MSG
7992	A63E	CD	40	00	CALL ZDSPMG	;PUT UP THE MESSAGE
7993	A641	.	.	.	; LOOP UNTIL 1-8 OR RETURN IS HIT	
7994	A641	.	.	.	TXS010 EQU \$	
7995	A641	CD	28	A4	CALL GETKEY	;GET A KEY
7996	A644	C2	41	A6	JNZ TXS010	;NONE HIT, CONTINUE LOOPING
7997	A647	FE	EF	.	CPI SFTCR	;SOFT RETURN?
7998	A649	CA	63	A6	JZ TXS015	;YES, EXIT
7999	A64C	FE	97	.	CPI SIZKEY	;SIZE KEY?
8000	A64E	CA	63	A6	JZ TXS015	;YES, EXIT
8001	A651	FE	31	.	CPI 61Q	;.GE.1?
8002	A653	DA	6C	A6	JC TXS020	;NO, BAD KEY
8003	A656	FE	39	.	CPI 71Q	;.GT. 8??
8004	A658	D2	6C	A6	JNC TXS020	;YES, BAD KEY
8005	A65B	.	.	.	; HAVE NEW SIZE 1-8	
8006	A65B	06	31	.	SUI 61Q	;WANT 0-7
8007	A65D	32	DA	FB	STA TXMAG	;STORE NEW SIZE
8008	A660	C3	25	A6	JMP TXS005	;DISPLAY NEW NEW SIZE
8009	A663	.	.	.	TXS015 EQU \$	
8010	A663	3A	DB	FB	LDA TANG	;COMPUTE NEW PARAMETES
8011	A666	CD	48	76	CALL ANGLE	
8012	A669	C3	43	00	JMP ZRSTDP	;RESTORE THE DISPLAY
8013	A66C	.	.	.	TXS020 EQU \$	
8014	A66C	CD	14	48	CALL ZBELL	;BAD KEY--BEEP
8015	A66F	C3	41	A6	JMP TXS010	;TRY AGAIN
8016	A672	.	.	.	SIZMSG EQU \$	
8017	A672	82	20	54	DB IVON,' TEXT SIZE (1-8) ',0	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 218
8019	A686	.	.	*****	
8020	A686	.	.	; TXTANG--SET TEXT ANGLE, SLANT FROM KEYBOARD	
8021	A686	.	.	*****	
8022	A686	.	.	TXTANG EQU \$	
8023	A686	CD	5C 69	CALL CKSCLD ;DONT CHANGE IF SCALED TEK 0	
8024	A689	C2	5B A7	JNZ TXT1 ;IGNORE IF SCALED TEK MODE	
8025	A68C	CD	42 76	CALL CKSLNT ;SAVE SLANT ON/OFF	
8026	A68F	F5	.	PUSH PSW	
8027	A690	AF	.	XRA A ;CLEAR ANY PENDING LABEL	
8028	A691	32	74 90	STA LBLCTR	
8029	A694	.	.	;PUT UP MESSAGE	
8030	A694	.	.	TXA005 EQU \$	
8031	A694	21	2F A7	LXI H,ANGMSG ;ANGLE MESSAGE	
8032	A697	22	F1 FF	SHLD ZMSGP1	
8033	A69A	21	0D FB	LXI H,NUMBUF ;WHERE CURRENT ANGLE WILL BE	
8034	A69D	22	EF FF	SHLD ZMSGP2	
8035	A6A0	.	.	; LOAD CURRENT ANGLE	
8036	A6A0	3A	DB FB	LDA TANG	
8037	A6A3	C6	31 .	ADI 610 ;CONVERT TO ASCII	
8038	A6A5	77	.	MOV M,A ;STORE IN BUFFER	
8039	A6A6	23	.	INX H	
8040	A6A7	36	20 .	MVI M,400 ;STORE TERMINATING SPACE	
8041	A6A9	23	.	INX H	
8042	A6AA	36	CC .	MVI M,EOL ;STORE END OF LINE	
8043	A6AC	23	.	INX H ;STORE END OF MESSAGE SEG	
8044	A6AD	36	00 .	MVI M,0	
8045	A6AF	.	.	; LOAD CURRENT SLANT	
8046	A6AF	23	.	INX H ;ASSUME SLANT IS OFF	
8047	A6B0	22	EB FF	SHLD ZMSGP4 ;STORE POINTER TO Y OR N	
8048	A6B3	36	4E .	MVI M,1160 ;STORE A CAP N	
8049	A6B5	CD	42 76	CALL CKSLNT ;IS SLANT REALLY OFF?	
8050	A6B8	CA	BD A6	JZ TXA010 ;YES, LEAVE THE N THERE	
8051	A6B8	36	59 .	MVI M,1310 ;NO, PUT IN A 'Y'	
8052	A6BD	.	.	TXA010 EQU \$	
8053	A6BD	23	.	INX H ;STORE TERMINATING SPACE	
8054	A6BE	36	20 .	MVI M,400	
8055	A6C0	23	.	INX H ;STORE THE EOP	
8056	A6C1	36	CE .	MVI M,ZEOP	
8057	A6C3	21	45 A7	LXI H,SLTMSG ;STORE THE FIXED PART OF MSG	
8058	A6C6	22	ED FF	SHLD ZMSGP3	
8059	A6C9	37	.	STC ;REPLACE DISPLAY WITH MSG	
8060	A6CA	CD	40 00	CALL ZDSPMG ;PUT UP MESSAGE	
8061	A6CD	.	.	TXA020 EQU \$	
8062	A6CD	CD	28 A4	CALL GETKEY ;GET A KEY	
8063	A6D0	C2	CD A6	JNZ TXA020 ;NONE HIT, LOOP	
8064	A6D3	FE	EF .	CPI SFTCR ;SOFT RETURN?	
8065	A6D5	CA	19 A7	JZ TXA050 ;YES, EXIT	
8066	A6D8	FE	96 .	CPI ANGKEY ;ANGLE KEY	
8067	A6DA	CA	19 A7	JZ TXA050 ;YES, EXIT	
8068	A6DD	.	.	; TEST FOR SLANT ON OR OFF	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 219
8069	A6DD	FE	59	.	CPI 131Q	;CAP 'Y'?
8070	A6DF	CA	09	A7	JZ TXA030	;YES, TURN SLANT ON
8071	A6E2	FE	79	.	CPI 171Q	;LOWER CASE 'Y'?
8072	A6E4	CA	09	A7	JZ TXA030	;YES, TURN SLANT ON
8073	A6E7	FE	4E	.	CPI 116Q	;CAP 'N'?
8074	A6E9	CA	11	A7	JZ TXA040	;YES, TURN SLANT OFF
8075	A6EC	FE	6E	.	CPI 156Q	;LOWER CASE 'N'?
8076	A6EE	CA	11	A7	JZ TXA040	;YES, TURN SLANT OFF
8077	A6F1	FE	35	.	CPI 65Q	;.GT. 4?
8078	A6F3	D2	03	A7	JNC TXA025	;YES, BEEP AND IGNORE
8079	A6F6	FE	31	.	CPI 61Q	;.GE. 1?
8080	A6F8	DA	03	A7	JC TXA025	;NO, BEEP AND IGNORE
8081	A6FB	.	.	.	; HAVE NEW ANGLE	
8082	A6FB	D6	31	.	SUI 61Q	;WANT 0-3
8083	A6FD	32	DB	FB	STA TANG	;STORE ANGLE
8084	A700	C3	94	A6	JMP TXA005	;UPDATE DISPLAY
8085	A703	.	.	.	TXA025 EQU \$	
8086	A703	CD	14	48	CALL ZBELL	;BEEP FOR BAD CHAR
8087	A706	C3	CD	A6	JMP TXA020	;LOOP
8088	A709	.	.	.	TXA030 EQU \$	
8089	A709	3E	01	.	MVI A,SLANT	;TURN SLANT ON
8090	A70B	CD	5A	A2	CALL STFLG6	
8091	A70E	C3	94	A6	JMP TXA005	;UPDATE DISPLAY
8092	A711	.	.	.	TXA040 EQU \$	
8093	A711	3E	01	.	MVI A,SLANT	;TURN SLANT OFF
8094	A713	CD	60	A2	CALL CLFLG6	
8095	A716	C3	94	A6	JMP TXA005	
8096	A719	.	.	.	TXA050 EQU \$	
8097	A719	.	.	.	; IF SLANT WAS ON BEFORE, AND WAS TURNED OFF,	
8098	A719	.	.	.	; DO A HT TO PREVENT UPRIGHT CHAR FROM OVERLAPPING	
8099	A719	CD	42	76	CALL CKSLNT	;FETCH CURRENT SLANT
8100	A71C	4F	.	.	MOV C,A	;LEAVE IN C
8101	A71D	F1	.	.	POP PSW	;FETCH OLD SLANT
8102	A71E	A9	.	.	XRA C	;DID IT CHANGE?
8103	A71F	CA	26	A7	JZ TXA060	;NO
8104	A722	A1	.	.	ANA C	;WAS SLANT TURNED OFF?
8105	A723	CC	9E	9D	CZ XHT	;IF YES, DO THE HT
8106	A726	.	.	.	TXA060 EQU \$	
8107	A726	3A	DB	FB	LDA TANG	;RECOMPUTE PARAMETERS
8108	A729	CD	48	76	CALL ANGLE	
8109	A72C	C3	43	00	JMP ZRSTDP	;RESTORE DISPLAY
8110	A72F	.	.	.	ANGMSG EQU \$	
8111	A72F	82	20	54	DB IVON,' TEXT ANGLE (1-4) ',0	
8112	A745	.	.	.	SLTMSG EQU \$	
8113	A745	82	20	53	DB IVON,' SLANTED? (Y OR N) ',0	
8114	A75B	.	.	.	;*****	
8115	A75B	.	.	.	; USER ATTEMPTED TO CHANGE TEXT PARAMETER WHILE	
8116	A75B	.	.	.	; IN SCALED TEK MODE. PUT UP MESSAGE	
8117	A75B	.	.	.	;*****	
8118	A75B	.	.	.	TXT1 EQU \$	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 220
=====
8119     A75B      21  7D  A7          LXI  H,TEKMSG  ;POINTER TO MESSAGE
8120     A75E      22  F1  FF          SHLD ZMSGP1
8121     A761      37  .   .           STC
8122     A762      CD  40  00          CALL ZDSPMG    ;PUT UP THE MESSAGE
8123     A765      .   .   .           TXT010 EQU $
8124     A765      .   .   .           ; WAIT FOR RETURN, TANG, OR TSIZ KEY
8125     A765      CD  28  A4          CALL GETKEY    ;GET A KEY
8126     A768      C2  65  A7          JNZ  TXT010    ;NONE READY, TRY AGAIN
8127     A76B      FE  EF  .           CPI  SFTCR     ;RETURN KEY?
8128     A76D      CA  43  00          JZ   ZRSTDP    ;YES, RESTORE DISPLAY
8129     A770      FE  97  .           CPI  SIZKEY    ;SIZE KEY?
8130     A772      CA  43  00          JZ   ZRSTDP    ;YES, RESTORE DISPLAY
8131     A775      FE  96  .           CPI  ANGKEY    ;ANGLE KEY?
8132     A777      CA  43  00          JZ   ZRSTDP    ;YES, RESTORE DISPLAY
8133     A77A      C3  65  A7          JMP  TXT010    ;TRY AGAIN
8134     A77D      .   .   .           TEKMSG EQU $
8135     A77D      82  49  4E          DB   IVON,'INVALID--SCALED MODE',ZEOP
=====
  
```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 221.
8137	A793	. . .	;*****	
8138	A793	. . .	; DISPLAY FUNCTION ROUTINES.	
8139	A793	. . .	; THE DRAW AND MOVE KEYS WILL ALSO BE EXECUTED	
8140	A793	. . .	;*****	
8141	A793	. . .	;*****	
8142	A793	. . .	; STOP KEY	
8143	A793	. . .	;*****	
8144	A793	. . .	DFSTOP EQU \$	
8145	A793	21 99 A7	LXI H,DF1 ;STOP MESSAGE	
8146	A796	C3 4D A9	JMP DFSEND	
8147	A799	. . .	DF1 EQU \$	
8148	A799	61 42 1B	DB 'AB',ESC,'*DT',EOL	
8149	A7A0	. . .	;*****	
8150	A7A0	. . .	; G CURSOR KEY	
8151	A7A0	. . .	;*****	
8152	A7A0	. . .	DFGC EQU \$;TOGGLE CURSOR	
8153	A7A0	21 B1 A7	LXI H,DF2 ;ASSUME CURSOR IS OFF NOW	
8154	A7A3	3A B0 90	LDA GFLGS3 ;MESSAGE TURNS CURSOR ON	
8155	A7A6	E6 80 .	ANI WANTGC ;REALLY OFF?	
8156	A7A8	CA 4D A9	JZ DFSEND ;YES	
8157	A7AB	21 B4 A7	LXI H,DF3 ;NO, USE OTHER STRING	
8158	A7AE	C3 4D A9	JMP DFSEND	
8159	A7B1	. . .	DF2 EQU \$	
8160	A7B1	64 4B CC	DB 'DK',EOL ;CURSOR ON	
8161	A7B4	. . .	DF3 EQU \$	
8162	A7B4	64 4C CC	DB 'DL',EOL ;CURSOR OFF	
8163	A7B7	. . .	;*****	
8164	A7B7	. . .	; RB LINE KEY	
8165	A7B7	. . .	;*****	
8166	A7B7	. . .	DFRB EQU \$;TOGGLE RBLINE	
8167	A7B7	21 C8 A7	LXI H,DF4 ;ASSUME OFF NOW	
8168	A7BA	3A B0 90	LDA GFLGS3	
8169	A7BD	E6 20 .	ANI WANTRB ;REALLY OFF?	
8170	A7BF	CA 4D A9	JZ DFSEND ;YES	
8171	A7C2	21 CB A7	LXI H,DF5 ;NO, TURN RBLIN OFF	
8172	A7C5	C3 4D A9	JMP DFSEND	
8173	A7C8	. . .	DF4 EQU \$	
8174	A7C8	64 4D CC	DB 'DM',EOL ;RBLINE ON	
8175	A7CB	. . .	DF5 EQU \$	
8176	A7CB	64 4E CC	DB 'DN',EOL ;RBLINE OFF	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 222
8178	A7CE	.	.	. ;*****	
8179	A7CE	.	.	. ; ZOOM KEY	
8180	A7CE	.	.	. ;*****	
8181	A7CE	.	.	. DFZM EQU \$;TOGGLE ZOOM	
8182	A7CE	21	DF A7	LXI H,DF6 ;ASSUME ZOOM OFF	
8183	A7D1	3A	AE 90	LDA GFLG55	
8184	A7D4	E6	02 .	ANI WANTZM ;ZOOM REALLY OFF?	
8185	A7D6	CA	4D A9	JZ DFSEND ;YES	
8186	A7D9	21	E2 A7	LXI H,DF7 ;NO	
8187	A7DC	C3	4D A9	JMP DFSEND	
8188	A7DF	.	.	. DF6 EQU \$	
8189	A7DF	64	47 CC	DB 'DG',EOL ;ZOOM ON	
8190	A7E2	.	.	. DF7 EQU \$	
8191	A7E2	64	48 CC	DB 'DH',EOL ;ZOOM OFF	
8192	A7E5	.	.	. ;*****	
8193	A7E5	.	.	. ; ZOOM IN KEY	
8194	A7E5	.	.	. ;*****	
8195	A7E5	.	.	. DFZIN EQU \$	
8196	A7E5	.	.	. ; INCREMENT CURRENT ZOOM SIZE AND DISPLAY	
8197	A7E5	3A	E1 FB	LDA MAG ;FETCH ZOOM SIZE (0-15)	
8198	A7E8	3C	.	. INR A ;CONVERT TO 1-16	
8199	A7E9	FE	10 .	CPI 16 ;AT MAX ALREADY?	
8200	A7EB	D2	02 A8	JNC DFZMSZ ;YES, DONT INCREMENT	
8201	A7EE	3C	.	. INR A ;NO, INCREMENT	
8202	A7EF	C3	02 A8	JMP DFZMSZ ;DISPLAY SIZE	
8203	A7F2	.	.	. ;*****	
8204	A7F2	.	.	. ; ZOOM OUT KEY	
8205	A7F2	.	.	. ;*****	
8206	A7F2	.	.	. DFZOUT EQU \$	
8207	A7F2	.	.	. ; DECREMENT ZOOM SIZE AND DISPLAY	
8208	A7F2	3A	E1 FB	LDA MAG ;FETCH SIZE (0-15)	
8209	A7F5	B7	.	. ORA A ;IS IT 0?	
8210	A7F6	C2	02 A8	JNZ DFZMSZ ;NO, DISPLAY SIZE-1	
8211	A7F9	3C	.	. INR A ;YES, MAKE SMALLEST = 1	
8212	A7FA	.	.	. ;*****	
8213	A7FA	.	.	. ; ROM BREAK 6	
8214	A7FA	C3	02 A8	JMP ZBRK6C	
8215	A7FD	.	.	. ORG ZBRK5+4000Q	
8216	A800	.	.	. ZBRK6 EQU \$	
8217	A800	54	.	. DB VERSN	
8218	A801	A8	.	. DB ZBRK6/256	
8219	A802	.	.	. ZBRK6C EQU \$	
8220	A802	.	.	. ;*****	
8221	A802	.	.	. DFZMSZ EQU \$	
8222	A802	.	.	. ; DISPLAY ZOOM SIZE IN A REG	
8223	A802	21	3F FB	LXI H,NUMBUF+50 ;BUFFER FOR ASCII	
8224	A805	22	EF FF	SHLD ZMSGP2	
8225	A808	36	64 .	MVI M,144Q ;STORE SMALL 0 FOR ESC SEQ	
8226	A80A	23	.	. INX H	
8227	A80B	CD	AB 00	CALL ZBNDCA ;CONVERT TO ASCII	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 223
=====
8228     A80E     36  49  .           MVI  M,111Q      ;CAP I => SET ZOOM SIZE
8229     A810     23  .   .           INX  H
8230     A811     36  FF  .           MVI  M,-1        ;END OF STRING
8231     A813     C3  50  A9          JMP  DFSND1
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 224
8233	A816	.	.	*****	
8234	A816	.	.	; CLEAR KEY	
8235	A816	.	.	*****	
8236	A816	.	.	DFCLR EQU \$;CLEAR SCREEN	
8237	A816	21	1C A8	LXI H,DF9 ;ASSUME IN HP MODE	
8238	A819	C3	4D A9	JMP DFSEND	
8239	A81C	.	.	DF9 EQU \$	
8240	A81C	64	41 CC	DB 'DA',EOL ;CLEAR THE SCREEN	
8241	A81F	.	.	*****	
8242	A81F	.	.	; G DISPLAY KEY	
8243	A81F	.	.	*****	
8244	A81F	.	.	DFGVD EQU \$;TOGGLE G VIDEO	
8245	A81F	21	30 A8	LXI H,DF11 ;ASSUME OFF	
8246	A822	3A	B5 90	LDA CURMOD ;REALLY OFF?	
8247	A825	E6	10 .	ANI GVENAB	
8248	A827	CA	4D A9	JZ DFSEND ;YES	
8249	A82A	21	33 A8	LXI H,DF12 ;NO	
8250	A82D	C3	4D A9	JMP DFSEND	
8251	A830	.	.	DF11 EQU \$	
8252	A830	64	43 CC	DB 'DC',EOL ;G VIDEO ON	
8253	A833	.	.	DF12 EQU \$	
8254	A833	64	44 CC	DB 'DD',EOL ;G VIDEO OFF	
8255	A836	.	.	*****	
8256	A836	.	.	; A/N DISPLAY KEY	
8257	A836	.	.	*****	
8258	A836	.	.	DFAVD EQU \$;TOGGLE A/N VIDEO	
8259	A836	21	47 A8	LXI H,DF13 ;ASSUME OFF	
8260	A839	3A	B2 90	LDA GFLGS1 ;REALLY OFF?	
8261	A83C	E6	20 .	ANI AVINH8	
8262	A83E	C2	4D A9	JNZ DFSEND ;YES	
8263	A841	21	4A A8	LXI H,DF14 ;NO	
8264	A844	C3	4D A9	JMP DFSEND	
8265	A847	.	.	DF13 EQU \$	
8266	A847	64	45 CC	DB 'DE',EOL ;A/N VIDEO ON	
8267	A84A	.	.	DF14 EQU \$	
8268	A84A	64	46 CC	DB 'DF',EOL ;A/N VIDEO OFF	
8269	A84D	.	.	*****	
8270	A84D	.	.	; AUTO PLOT MENU KEY	
8271	A84D	.	.	*****	
8272	A84D	.	.	DFMENU EQU \$	
8273	A84D	21	53 A8	LXI H,DF15 ;TURN MENU ON	
8274	A850	C3	4D A9	JMP DFSEND	
8275	A853	.	.	DF15 EQU \$	
8276	A853	61	46 CC	DB 'AF',EOL ;AUTO PLOT MENU ON	
8277	A856	.	.	*****	
8278	A856	.	.	; AUTO PLOT KEY	
8279	A856	.	.	*****	
8280	A856	.	.	DFAPON EQU \$;TURN AUTO PLOT UN	
8281	A856	21	5C A8	LXI H,DF16	
8282	A859	C3	4D A9	JMP DFSEND	

13255
2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                PAGE 225
=====
8283     A85C      .      .      .      DF16  EQU  $
8284     A85C      61     41     CC      DB   'AA',EOL ;AUTO PLOT ON
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 226
=====
8286     A85F      . . .      ;*****
8287     A85F      . . .      ; AUTO PLOT AXES KEY
8288     A85F      . . .      ;*****
8289     A85F      . . .      DFAXES EQU $ ;DRAW AXES
8290     A85F      21 65 A8      LXI H,DF17
8291     A862      C3 4D A9      JMP DFSEND
8292     A865      . . .      DF17 EQU $
8293     A865      61 43 CC      DB 'AC',EOL ;DRAW AXES
8294     A868      . . .      ;*****
8295     A868      . . .      ; GRAPHICS TEXT KEY
8296     A868      . . .      ;*****
8297     A868      . . .      DFSTX EQU $ ;START GRAPHICS TEXT
8298     A868      21 78 A8      LXI H,DF18 ;SEND TEXT, CURSOR ON
8299     A86B      CD 4D A9      CALL DFSEND
8300     A86E      CD 5C 69      CALL CKSCLD ;DONT CHANGE MODE IF SCLD TE
8301     A871      C0 . .      RNZ ;YES, DONT CHANGE MODE
8302     A872      21 7F A8      LXI H,DF18A ;NO, SET JAM PATTERN
8303     A875      C3 4D A9      JMP DFSEND
8304     A878      . . .      DF18 EQU $
8305     A878      64 53 1B      DB 'DS',ESC,'*DK',EOL ;GTEXT, CURSOR ON
8306     A87F      . . .      DF18A EQU $
8307     A87F      6D 34 41      DB 'M4A',EOL ;JAM PATTERN ON
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 227
8309	A883	.	.	*****	
8310	A883	.	.	; TEXT ANGLE KEY	
8311	A883	.	.	*****	
8312	A883	.	.	DFTANG EQU \$	
8313	A883	.	.	; DISPLAY CURRENT ANGLE AND SLANT	
8314	A883	3A	DB FB	LDA TANG ;FETCH CURRENT ANGLE	
8315	A886	C6	31 .	ADI 61Q ;CONVERT TO ASCII	
8316	A888	21	3F FB	LXI H,NUMBUF+50 ;MESSAGE BUFFER	
8317	A888	22	EF FF	SHLD ZMSGP2	
8318	A88E	36	6D .	MVI M,155Q ;STORE SMALL M FOR ESC SEQ	
8319	A890	23	.	INX H	
8320	A891	77	.	MOV M,A ;STORE ANGLE	
8321	A892	23	.	INX H	
8322	A893	36	4E .	MVI M,116Q ;CAP N => SET TEXT ANGLE	
8323	A895	23	.	INX H	
8324	A896	36	FF .	MVI M,-1 ;SEND END OF MESSAGE	
8325	A898	CD	50 A9	CALL DFSND1 ;SEND TEXT ANGLE	
8326	A89B	21	AA A8	LXI H,DF19A ;ASSUME SLANT OFF	
8327	A89E	CD	42 76	CALL CKSLNT ;IS SLANT REALLY OFF?	
8328	A8A1	CA	4D A9	JZ DFSEND ;YES	
8329	A8A4	21	AD A8	LXI H,DF19B ;NO	
8330	A8A7	C3	4D A9	JMP DFSEND	
8331	A8AA	.	.	DF19A EQU \$	
8332	A8AA	6D	4F CC	DB 'MO',EOL ;SLANT ON	
8333	A8AD	.	.	DF19B EQU \$	
8334	A8AD	6D	50 CC	DB 'MP',EOL ;SLANT OFF	
8335	A8B0	.	.	*****	
8336	A8B0	.	.	; TEXT SIZE KEY	
8337	A8B0	.	.	*****	
8338	A8B0	.	.	DFTSIZ EQU \$	
8339	A8B0	.	.	; DISPLAY CURRENT SIZE	
8340	A8B0	3A	DA FB	LDA TXMAG ;FETCH CURRENT SIZE	
8341	A8B3	C6	31 .	ADI 61Q ;CONVERT TO ASCII	
8342	A8B5	21	3F FB	LXI H,NUMBUF+50 ;BUFFER	
8343	A8B8	22	EF FF	SHLD ZMSGP2	
8344	A8BB	36	6D .	MVI M,155Q ;STORE SMALL M FOR ESC SEQ	
8345	A8BD	23	.	INX H	
8346	A8BE	77	.	MOV M,A ;STORE SIZE	
8347	A8BF	23	.	INX H	
8348	A8C0	36	4D .	MVI M,115Q ;CAP M=> SET TEXT SIZE	
8349	A8C2	23	.	INX H	
8350	A8C3	36	FF .	MVI M,-1 ;END OF STRING	
8351	A8C5	C3	50 A9	JMP DFSND1	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 228
8353	A8C8	.	.	.	;*****
8354	A8C8	.	.	.	; DFMOV--GENERATE STRING TO MOVE TO CURSOR, AND
8355	A8C8	.	.	.	; ALSO EXECUTE MOVE IF NOT IN SEND CONTROL CODES
8356	A8C8	.	.	.	;*****
8357	A8C8	.	.	.	DFMOV EQU \$
8358	A8C8	.	.	.	; DONT EXECUTE IF CURSOR OFF
8359	A8C8	CD	81	A9	CALL GCCHK ;CURSOR OFF?
8360	A8CB	C0	.	.	RNZ ;YES, IGNORE
8361	A8CC	21	F6	A8	LXI H,DF20 ;INITIAL PREAMBLE
8362	A8CF	22	EF	FF	SHLD ZMSGP2
8363	A8D2	21	3F	FB	LXI H,NUMBUF+50 ;BUFFER
8364	A8D5	22	ED	FF	SHLD ZMSGP3
8365	A8D8	CD	F9	A8	CALL DFM1 ;GET CURSOR POSTION
8366	A8DB	36	5A	.	MVI M,132Q ;CAP Z => END ESC SEQ
8367	A8DD	23	.	.	INX H
8368	A8DE	36	FF	.	MVI M,-1 ;END OF STRING
8369	A8E0	CD	50	A9	CALL DFSND1 ;DISPLAY THE STRING
8370	A8E3	.	.	.	; IF NOT IN SEND FUNCTIONS, EXECUTE THE MOVE
8371	A8E3	3A	FB	FF	LDA KBJMP1 ;CHECK STRAP A
8372	A8E6	E6	01	.	ANI AJMPR
8373	A8E8	C0	.	.	RNZ
8374	A8E9	CD	C6	00	CALL ZCHKSF ;DONT EXECUTE, STRAP OUT
8375	A8EC	C0	.	.	RNZ ;SOFT KEYS UP?
8376	A8ED	CD	D2	00	CALL ZCKRMT ;YES, EXIT NOW
8377	A8F0	CC	89	A9	CZ CRLF ;IN REMOTE?
8378	A8F3	C3	EA	A5	JMP KBMOVE ;NO, DO A RETURN, LINE FEED
8379	A8F6	.	.	.	DF20 EQU \$;DO THE MOVE
8380	A8F6	70	61	00	DB 'PA',0 ;ASCII MOVE
8381	A8F9	.	.	.	;
8382	A8F9	.	.	.	DFM1 EQU \$
8383	A8F9	.	.	.	; GET CURSOR POSITION AND PUT INTO BUFFER
8384	A8F9	.	.	.	; HL = BUFFER
8385	A8F9	EB	.	.	XCHG ;DE = BUFFER
8386	A8FA	2A	CF	90	LHLD NEWGCX ;X COORD
8387	A8FD	CD	71	A9	CALL GETVAL ;GET ASCII COORD
8388	A900	36	2C	.	MVI M,54Q ;PUT COMMA BETWEEN X AND Y
8389	A902	23	.	.	INX H
8390	A903	EB	.	.	XCHG ;DE = BUFFER
8391	A904	2A	CD	90	LHLD NEWGCY ;Y COORD
8392	A907	CD	71	A9	CALL GETVAL ;CONVERT TO ASCII
8393	A90A	C9	.	.	RET ;HL = BUFFER POINTER

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 229
8395	A90B	.	.	.	;*****
8396	A90B	.	.	.	; DFDRAW--
8397	A90B	.	.	.	; PUT UP ESC SEQ TO DRAW VECTOR FROM CURRENT COORD
8398	A90B	.	.	.	; TO CURSOR POSITION. IF NOT IN SEND CONTROL CODE,
8399	A90B	.	.	.	; ALSO DO THE DRAW
8400	A90B	.	.	.	;*****
8401	A90B	.	.	.	DFDRAW EQU \$
8402	A90B	CD	81	A9	CALL GCCHK ;CURSOR OFF?
8403	A90E	C0	.	.	RNZ ;YES, IGNORE
8404	A90F	21	F6	A8	LXI H,DF20 ;PLOT PREAMBLE
8405	A912	22	EF	FF	SHLD ZMSGP2
8406	A915	21	3F	FB	LXI H,NUMBUF+50 ;BUFFER FOR ASCII
8407	A918	22	ED	FF	SHLD ZMSGP3
8408	A91B	EB	.	.	XCHG ;DE = BUFFER POINTER
8409	A91C	2A	DE	90	LHLD XCURR ;CONVERT CURRENT X COORD
8410	A91F	CD	71	A9	CALL GETVAL
8411	A922	36	2C	.	MVI M,540 ;SEPARATED BY COMMA
8412	A924	23	.	.	INX H
8413	A925	EB	.	.	XCHG ;DE = BUFFER POINTER
8414	A926	2A	DC	90	LHLD YCURR ;CONVERT Y COORD
8415	A929	CD	71	A9	CALL GETVAL
8416	A92C	36	2C	.	MVI M,540 ;SEPARATED BY COMMA
8417	A92E	.	.	.	; HAVE SEQUENCE TO MOVE TO CURRENT POINT
8418	A92E	23	.	.	INX H
8419	A92F	CD	F9	A8	CALL DFM1 ;CONVERT CURSOR COORDS
8420	A932	36	5A	.	MVI M,1320 ;CAP Z => END OF ESC SEQ
8421	A934	23	.	.	INX H
8422	A935	36	FF	.	MVI M,-1 ;END OF STRING
8423	A937	CD	50	A9	CALL DFSND1 ;DISPLAY THE STRING
8424	A93A	.	.	.	; IF NOT IN SEND FUNC CODES MODE, EXECUTE THE DRW
8425	A93A	3A	FB	FF	LDA KBJMP1
8426	A93D	E6	01	.	ANI AJMPR ;STRAP A OUT?
8427	A93F	C0	.	.	RNZ ;YES, DONT EXECUTE
8428	A940	CD	C6	00	CALL ZCHKSF ;SOFT KEYS UP?
8429	A943	C0	.	.	RNZ ;YES, EXIT NOW
8430	A944	CD	D2	00	CALL ZCKRMT ;IN REMOTE?
8431	A947	CC	89	A9	CZ CRLF ;NO, DO A CR/LF
8432	A94A	C3	CE	A5	JMP KBDRAW ;DO THE DRAW

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 230
8434	A94D	.	.	*****	
8435	A94D	.	.	; DFSEND--SEND STRING TO LOCLIN ROUTINE TO	
8436	A94D	.	.	; EITHER PUT ON DISPLAY OR SEND TO DATACOM	
8437	A94D	.	.	; ENTRY TO DFSEND--HL = POINTER TO ST OF STRING	
8438	A94D	.	.	*****	
8439	A94D	.	.	DFSEND EQU \$	
8440	A94D	22	EF FF	SHLD ZMSGP2 ;STORE STRING POINTER	
8441	A950	.	.	DFSND1 EQU \$	
8442	A950	21	6E A9	LXI H,ESCSTR ;INITIAL ESC * STRING	
8443	A953	22	F1 FF	SHLD ZMSGP1	
8444	A956	21	F2 FF	LXI H,ZMSGP1+1	
8445	A959	.	.	DFS010 EQU \$	
8446	A959	56	.	MOV D,M ;GET STRING POINTER	
8447	A95A	2B	.	DCX H	
8448	A95B	5E	.	MOV E,M	
8449	A95C	2B	.	DCX H	
8450	A95D	.	.	; DE = POINTER TO CURRENT CHAR	
8451	A95D	.	.	DFS020 EQU \$	
8452	A95D	1A	.	LDAX D ;FETCH THE CHAR	
8453	A95E	B7	.	ORA A ;END OF STRING?	
8454	A95F	F8	.	RM ;YES, DONE	
8455	A960	CA	59 A9	JZ DFS010 ;NULL=>GO TO NEXT STRING	
8456	A963	E5	.	PUSH H ;SAVE POINTERS	
8457	A964	D5	.	PUSH D	
8458	A965	CD	E1 00	CALL ZLCLN2 ;PROCESS THE CHAR	
8459	A968	D1	.	POP D	
8460	A969	13	.	INX D ;GET NEXT CHAR	
8461	A96A	E1	.	POP H	
8462	A96B	C3	5D A9	JMP DFS020 ;DO THE NEXT CHAR	
8463	A96E	.	.	ESCSTR EQU \$	
8464	A96E	1B	2A 00	DB ESC,'*',0	
8465	A971	.	.	*****	
8466	A971	.	.	; GETVAL--CONVERT HL TO ASCII AND PUT INTO BUFFER	
8467	A971	.	.	; ENTRY HL = VALUE	
8468	A971	.	.	; DE = BUFFER POINTER	
8469	A971	.	.	*****	
8470	A971	.	.	GETVAL EQU \$	
8471	A971	7C	.	MOV A,H ;IS VALUE +?	
8472	A972	B7	.	ORA A	
8473	A973	F2	7D A9	JP GTV010 ;YES	
8474	A976	3E	2D .	MVI A,550 ;STORE A - SIGN	
8475	A978	12	.	STAX D	
8476	A979	13	.	INX D ;UPDATE BUFFER POINTER	
8477	A97A	CD	09 A3	CALL NEGATE ;GET ABS VAL	
8478	A97D	.	.	GTV010 EQU \$	
8479	A97D	EB	.	XCHG ;HL = POINTER, DE = VALUE	
8480	A97E	C3	A8 00	JMP ZBNDEC ;DO THE ASCII CONVERSION	
8481	A981	.	.	*****	
8482	A981	.	.	; GCCHK--SEE IF GRAPHICS CURSOR ON OR NOT	
8483	A981	.	.	; EXIT NZ => OFF	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 231
=====
8484     A981      . . .      ;*****
8485     A981      . . .      GCCHK EQU $
8486     A981      3A 80 90    LDA GFLGS3 ;GET CURSOR FLAGS
8487     A984      E6 86 .     ANI WANTGC+SUPR1+SUPR2 ;CURSOR OFF OR
8488     A986      FE 80 .     CPI WANTGC ;SUPRESSED?
8489     A988      C9 . .     RET
8490     A989      . . .      ;*****
8491     A989      . . .      ; CRLF--DO A CARRIAGE RETURN, LINE FEED
8492     A989      . . .      ;*****
8493     A989      . . .      CRLF EQU $
8494     A989      CD C0 00    CALL ZCRRET ;DO THE RETURN
8495     A98C      C3 8B 00    JMP ZLNFD ;DO THE LINE FEED
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 232
8497	A98F	.	.	*****	
8498	A98F	.	.	; START OF AUTO PLOT CODE	
8499	A98F	.	.	*****	
8500	A98F	.	.	*****	
8501	A98F	.	.	; APSEQ--ESC * A RECEIVED, SET UP FOR AUTO PLOT	
8502	A98F	.	.	*****	
8503	A98F	.	.	APSEQ EQU \$	
8504	A98F	21	68 62	LXI H,APTAB ;LOAD AUTO PLOT RANGE TABLE	
8505	A992	22	D2 FF	SHLD ZRNGTA	
8506	A995	3E	50 .	MVI A,WANTAX+WANTAP ;CLEAR ESC SEQ FLAGS	
8507	A997	CD	C2 B8	CALL CLAPP2	
8508	A99A	.	.	; LOAD PARAMETER BUFFER WITH SPACES	
8509	A99A	.	.	CLRBUF EQU \$	
8510	A99A	21	B9 90	LXI H,PRMBUF ;BASE OF BUFFER	
8511	A99D	3E	0F .	MVI A,15 ;15 LOCATIONS TO CLEAR	
8512	A99F	.	.	CLB010 EQU \$	
8513	A99F	36	20 .	MVI M,400 ;STORE A SPACE	
8514	A9A1	23	. .	INX H	
8515	A9A2	3D	. .	DCR A ;ALL DONE?	
8516	A9A3	C2	9F A9	JNZ CLB010 ;NO, KEEP LOOPING	
8517	A9A6	32	B6 90	STA PRMDEX ;CLEAR PARAMETER COUNT	
8518	A9A9	C3	82 B8	JMP APLTOF ;TURN AUTO PLOT OFF	
8519	A9AC	.	.	*****	
8520	A9AC	.	.	; PUTBUF--PARAMETER RECEIVED, PUT INTO BUFFER	
8521	A9AC	.	.	; VALID AUTO PLOT PARAMETERS ARE 0-9,+,-,.,E	
8522	A9AC	.	.	*****	
8523	A9AC	.	.	PUTBUF EQU \$	
8524	A9AC	21	B6 90	LXI H,PRMDEX ;PARAMETER COUNT	
8525	A9AF	7E	. .	MOV A,M	
8526	A9B0	FE	10 .	CPI MAXAP ;HAVE MAX NO. OF PARAMS?	
8527	A9B2	D0	. .	RNC ;YES, IGNORE	
8528	A9B3	34	. .	INR M ;NO, UPDATE COUNT	
8529	A9B4	5F	. .	MOV E,A	
8530	A9B5	16	00 .	MVI D,0 ;INDEX TO SLOT IN PRMBUF	
8531	A9B7	21	B9 90	LXI H,PRMBUF ;BASE OF BUFFER	
8532	A9BA	19	. .	DAD D	
8533	A9BB	3A	88 FF	LDA ZCHAR ;FETCH THE PARAMETER	
8534	A9BE	77	. .	MOV M,A ;STORE IT	
8535	A9BF	C9	. .	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 233
8537	A9C0	.	.	*****	
8538	A9C0	.	.	; XFRBUF--COMMAND (H-W) HAS BEEN RECEIVED.	
8539	A9C0	.	.	; TRANSFER CONTENTS OF PRMBUF TO PROPER AUTO PLOT	
8540	A9C0	.	.	; FIELD IN DISPLAY MEMORY, AND CLOSE THE FIELD.	
8541	A9C0	.	.	*****	
8542	A9C0	.	.	XFRBUF EQU \$	
8543	A9C0	3A	88 FF	LDA ZCHAR ;FETCH COMMAND = MENU FIELD	
8544	A9C3	E6	1F .	ANI 37Q ;DELETE UPPER/LOWER CASE	
8545	A9C5	D6	08 .	SUI APOFST ;CONVERT TO 0-15	
8546	A9C7	CD	CD A9	CALL XFRBF1 ;LOAD MENU	
8547	A9CA	C3	4D AA	JMP APEXIT	
8548	A9CD	.	.	XFRBF1 EQU \$;(INTERNAL ENTRY)	
8549	A9CD	.	.	; FIND LOCATION IN MEMORY OF CURRENT FIELD	
8550	A9CD	CD	ED AD	CALL PUTFLD	
8551	A9D0	.	.	; NOW XFER FROM PRMBUF TO MENU	
8552	A9D0	2A	FC FA	LHLD DSPFLD ;ADDRESS OF MENU FIELD	
8553	A9D3	11	B9 90	LXI D,PRMBUF ;SOURCE OF DATA	
8554	A9D6	.	.	; XFER UNTIL EOL OR IVOFF CODE IS FOUND IN MENU	
8555	A9D6	.	.	XFB010 EQU \$	
8556	A9D6	7E	.	MOV A,M ;FETCH DATA IN MENU	
8557	A9D7	B7	.	ORA A ;END OF FIELD?	
8558	A9D8	FA	E2 A9	JM XFB020 ;YES, CLOSE FIELD, EXIT	
8559	A9DB	1A	.	LDAX D ;FETCH PARAMETER	
8560	A9DC	77	.	MOV M,A ;STORE IN MENU	
8561	A9DD	2B	.	DCX H ;UPDATE POINTERS	
8562	A9DE	13	.	INX D	
8563	A9DF	C3	D6 A9	JMP XFB010	
8564	A9E2	.	.	XFB020 EQU \$	
8565	A9E2	CD	C7 AE	CALL CLOSE ;CLOSE THE FIELD	
8566	A9E5	C3	4D AA	JMP APEXIT	
8567	A9E8	.	.	*****	
8568	A9E8	.	.	; GUAP--START AUTO PLOT FROM ESCAPE SEQUENCE	
8569	A9E8	.	.	; IT IS NOT ACTUALLY TURNED ON UNTIL ESCAPE	
8570	A9E8	.	.	; SEQUENCE TERMINATES	
8571	A9E8	.	.	*****	
8572	A9E8	.	.	GOAP EQU \$	
8573	A9E8	3E	10 .	MVI A,WANTAP ;SET FLAG TO TURN AP ON AT	
8574	A9EA	CD	BC B8	CALL STAPF2 ;END OF ESC SEQ	
8575	A9ED	C3	4D AA	JMP APEXIT	
8576	A9F0	.	.	*****	
8577	A9F0	.	.	; STOPAP--TURN AUTO PLOT OFF FROM ESCAPE SEQ	
8578	A9F0	.	.	*****	
8579	A9F0	.	.	STOPAP EQU \$	
8580	A9F0	3E	10 .	MVI A,WANTAP ;CLEAR 'TURN AP ON' FLAG	
8581	A9F2	CD	C2 B8	CALL CLAPF2	
8582	A9F5	C3	4D AA	JMP APEXIT	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 234
8584	A9F8	.	.	*****	
8585	A9F8	.	.	; DWAXES--DRAW AUTO PLOT AXES. THEY ARE NOT	
8586	A9F8	.	.	; ACTUALLY DRAWN UNTIL THE ESC SEQ TERMINATES	
8587	A9F8	.	.	*****	
8588	A9F8	.	.	DWAXES EQU \$	
8589	A9F8	3E	40	MVI A,WANTAX ;SET FLAG TO DRAW AXES AT	
8590	A9FA	CD	BC	CALL STAFF2 ;END OF ESC SEQ	
8591	A9FD	C3	4D	AA JMP APEXIT	
8592	AA00	.	.	*****	
8593	AA00	.	.	; CLRMNU--CLEAR ALL MENU FIELDS	
8594	AA00	.	.	*****	
8595	AA00	.	.	CLRMNU EQU \$	
8596	AA00	CD	9A	A9 CALL CLRBUF ;CLEAR PARAMETER BUFFER	
8597	AA03	21	02	FB LXI H,MUFLD ;START WITH FIRST FIELD	
8598	AA06	36	FF	. MVI M,-1	
8599	AA08	.	.	CLM010 EQU \$	
8600	AA08	3A	02	FB LDA MUFLD ;FETCH FIELD	
8601	AA0B	3C	.	. INR A	
8602	AA0C	FE	10	. CPI BOTFLD+1 ;LAST FIELD?	
8603	AA0E	D2	17	AA JNC CLM020 ;YES, DONE	
8604	AA11	CD	CD	A9 CALL XFRBF1 ;NO, CLEAR IT	
8605	AA14	C3	08	AA JMP CLM010 ;DO THE NEXT ONE	
8606	AA17	.	.	CLM020 EQU \$	
8607	AA17	CD	9B	AD CALL HOME ;HOME THE CURSOR	
8608	AA1A	C3	4D	AA JMP APEXIT	
8609	AA1D	.	.	*****	
8610	AA1D	.	.	; DSPMNU -- TURN AUTO PLOT MENU ON FROM ESC SEQ	
8611	AA1D	.	.	*****	
8612	AA1D	.	.	DSPMNU EQU \$	
8613	AA1D	CD	EE	B8 CALL MUCHK ;MENU ALREADY UP?	
8614	AA20	C2	4D	AA JNZ APEXIT ;YES, DONE	
8615	AA23	CD	E3	AA CALL MUON ;TURN THE MENU ON	
8616	AA26	C3	4D	AA JMP APEXIT	
8617	AA29	.	.	*****	
8618	AA29	.	.	; OFFMNU--TURN MENU OFF FROM ESC SEQ	
8619	AA29	.	.	*****	
8620	AA29	.	.	OFFMNU EQU \$	
8621	AA29	CD	EE	B8 CALL MUCHK ;ALREADY OFF?	
8622	AA2C	CA	C1	99 JZ GEXIT ;YES, DONE	
8623	AA2F	CD	08	AB CALL MUOFF ;NO, TURN IT OFF	
8624	AA32	C3	4D	AA JMP APEXIT	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 235
8626	AA35	.	.	*****	
8627	AA35	.	.	; APESC--ESC RECEIVED IN ESC * A SEQ. IF MENU	
8628	AA35	.	.	; UP, USE MENU RANGE TABLE, ELSE PROCESS ESC	
8629	AA35	.	.	; NORMALLY	
8630	AA35	.	.	*****	
8631	AA35	.	.	APESC EQU \$	
8632	AA35	CD	EE B8	CALL MUCHK ;IS THE MENU UP?	
8633	AA38	CA	B7 00	JZ ZESCAP ;NO, PROCESS ESCAPE NORMALLY	
8634	AA38	C3	B9 AD	JMP MUESC ;YES, USE MENU RANGE TABLE	
8635	AA3E	.	.	*****	
8636	AA3E	.	.	; MUABT--ESC * <NOT A> RECEIVED WHILE MENU UP	
8637	AA3E	.	.	; TURN THE MENU OFF AND PROCESS NORMALLY	
8638	AA3E	.	.	*****	
8639	AA3E	.	.	MUABT EQU \$	
8640	AA3E	CD	08 AB	CALL MUOFF ;TURN THE MENU OFF	
8641	AA41	21	76 60	LXI H,GTAB ;RESTORE ESC * RANGE TABLE	
8642	AA44	C3	37 6A	JMP TKRPT1 ;REPEAT CHAR WITH NEW TABLE	
8643	AA47	.	.	*****	
8644	AA47	.	.	; PRMABT--ESC & RECEIVED WHILE MENU UP. TURN	
8645	AA47	.	.	; MENU OFF AND PROCESS NORMALLY	
8646	AA47	.	.	*****	
8647	AA47	.	.	PRMABT EQU \$	
8648	AA47	CD	08 AB	CALL MUOFF ;TURN THE MENU OFF]	
8649	AA4A	C3	DE 00	JMP ZPRMSQ ;SET UP FOR ESC & SEQ	
8650	AA4D	.	.	*****	
8651	AA4D	.	.	; APEXIT--TERMINATE AUTO PLOT ESCAPE SEQUENCE	
8652	AA4D	.	.	*****	
8653	AA4D	.	.	APEXIT EQU \$	
8654	AA4D	CD	9A A9	CALL CLRBUF ;CLEAR PARAMETER BUFFER	
8655	AA50	3A	88 FF	LDA ZCHAR ;FETCH COMMAND	
8656	AA53	E6	20 .	ANI LWRCS ;WAS IT LOWER CASE?	
8657	AA55	C2	C1 99	JNZ GEXIT ;YES, DONT TERNINATE SEQ	
8658	AA58	.	.	; TERMNINATE SEQUENCE--IF USER WANTS AUTO PLOT ON	
8659	AA58	.	.	; OR AXES DRAWN, DO IT NOW	
8660	AA58	3A	97 FB	LDA APFLG2 ;DRAW AXES?	
8661	AA5B	E6	40 .	ANI WANTAX	
8662	AA5D	C4	EF B1	CNZ APAXES ;DRAW THEM IF YES	
8663	AA60	3A	97 FB	LDA APFLG2 ;TURN AUTO PLOT ON?	
8664	AA63	E6	10 .	ANI WANTAP	
8665	AA65	C4	99 B7	CNZ APLTON ;TURN IT ON IF YES	
8666	AA68	3E	50 .	MVI A,WANTAX+WANTAP ;CLEAR FLAGS	
8667	AA6A	CD	C2 B8	CALL CLAPF2	
8668	AA6D	CD	C1 99	CALL GEXIT	
8669	AA70	.	.	; DISABLE CR/LF (FOR MENU LOADING SEQUENCES)	
8670	AA70	C3	C5 75	JMP NOCRLF	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 236
=====
8672     AA73     . . .      ;*****
8673     AA73     . . .      ; APINIT--COMPUTE SCALE FACTORS FOR AUTO PLOT
8674     AA73     . . .      ;*****
8675     AA73     . . .      APINIT EQU $
8676     AA73     . . .      ; XSCALE = XAXLEN/(XMAX-XMIN)
8677     AA73     . . .      ; YSCALE = YAXLEN/(YMAX-YMIN)
8678     AA73     . . .      ; AXLEN IS REFERED TO AS FP719, FP359 ALTHOUGH
8679     AA73     . . .      ; THEY ARE ACTUALLY SMALLER DUE TO LABELS
8680     AA73     . . .      ; COMPUTE X SCALE FACTOR
8681     AA73     21 B8 FB      LXI H,XMAXBF ;COMPUTE MAX - MIN
8682     AA76     CD 6E BD      CALL LOD
8683     AA79     21 8C FB      LXI H,XMINBF
8684     AA7C     CD D3 BD      CALL SB
8685     AA7F     CD 50 BD      CALL ABS ;INSURE +
8686     AA82     21 A2 90     LXI H,XSCALE ;STORE MAX-MIN
8687     AA85     CD 3E BD      CALL STR
8688     AA88     21 C1 AA      LXI H,FP719 ;LOAD AXIS LENGTH
8689     AA8B     CD 6E BD      CALL LOD
8690     AA8E     21 A2 90     LXI H,XSCALE ;RECALL MAX-MIN
8691     AA91     CD 84 BD      CALL DIV ;DIVIDE INTO AXLEN
8692     AA94     21 A2 90     LXI H,XSCALE
8693     AA97     CD 3E BD      CALL STR ;STORE X SCALE FACTOR
8694     AA9A     . . .      ; COMPUTE Y SCALE FACTOR
8695     AA9A     21 80 FB      LXI H,YMAXBF ;COMPUTE MAX-MIN
8696     AA9D     CD 6E BD      CALL LOD
8697     AAA0     21 84 FB      LXI H,YMINBF
8698     AAA3     CD 03 BD      CALL SB
8699     AAA6     CD 50 BD      CALL ABS ;INSURE +
8700     AAA9     21 9E 90     LXI H,YSCALE
8701     AAAC     CD 3E BD      CALL STR ;SAVE MAX-MIN
8702     AAAF     21 C5 AA      LXI H,FP359 ;LOAD AXIS LENGTH
8703     AAB2     CD 6E BD      CALL LOD
8704     AAB5     21 9E 90     LXI H,YSCALE ;RECALL MAX-MIN
8705     AAB8     CD 84 BD      CALL DIV ;DIVIDE INTO AX LEN
8706     AABB     21 9E 90     LXI H,YSCALE ;STORE Y SCALE FACTOR
8707     AABE     C3 3E BD      JMP STR
8708     AAC1     . . .      ;
8709     AAC1     8A 19 80     FP719 DB 212Q,31Q,200Q,0Q ;REALLY 614
8710     AAC5     89 18 00     FP359 DB 211Q,30Q,0Q,0Q ;REALLY 304
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 237
8712	AAC9	.	.	*****	
8713	AAC9	.	.	; APMUON--TURN AUTO PLOT MENU ON	
8714	AAC9	.	.	*****	
8715	AAC9	.	.	APMUON EQU \$	
8716	AAC9	CD	EE B8	CALL MUCHK ;MENU ALREADY ON?	
8717	AACC	C0	.	RNZ ;YES	
8718	AACD	CD	82 B8	CALL APLTOF ;TURN AUTO PLOT OFF	
8719	AAD0	CD	E3 AA	CALL MUON ;TURN THE MENU ON	
8720	AAD3	21	F8 61	LXI H,MUTB ;SET RANGE TABLE	
8721	AAD6	C3	86 63	JMP SETRTB	
8722	AAD9	.	.	*****	
8723	AAD9	.	.	; APMUOF--TURN AUTO PLOT MENU OFF	
8724	AAD9	.	.	; ENTRY--DONT CARE	
8725	AAD9	.	.	; EXIT---ALL REGISTERS DESTROYED	
8726	AAD9	.	.	*****	
8727	AAD9	.	.	APMUOF EQU \$	
8728	AAD9	CD	EE B8	CALL MUCHK ;MENU ALREADY OFF?	
8729	AADC	C8	.	RZ ;YES	
8730	AADD	CD	08 AB	CALL MUOFF ;RESTORE NORMAL DISPLAY	
8731	AAE0	.	.	;	
8732	AAE0	C3	4F 00	JMP ZESCND ;RESTORE RANGE TABLES	
8733	AAE3	.	.	;	
8734	AAE3	.	.	*****	
8735	AAE3	.	.	; MUON--PUT AP MENU ON DISPLAY	
8736	AAE3	.	.	*****	
8737	AAE3	.	.	MUON EQU \$	
8738	AAE3	CD	31 A4	CALL VIDEO1 ;SUPPRESS GRAPHICS	
8739	AAE6	3E	01 .	MVI A,MENUON ;SET MENU FLAG	
8740	AAE8	CD	BC B8	CALL STAPF2	
8741	AAEB	2A	CB FF	LHLD ZTOPLN ;SAVE TOP LINE OF CURRENT	
8742	AAEE	22	00 FB	SHLD TOPSAV ;DISPLAY	
8743	AAF1	2A	C0 FF	LHLD ZCURROW ;SAVE CURSOR POSITION	
8744	AAF4	22	06 FB	SHLD CURSAV ;OF CURRENT DISPLAY	
8745	AAF7	21	DA FA	LXI H,MUBUF ;START OF MENU	
8746	AAFA	22	FE FF	SHLD ZDSPST ;START DMA THERE	
8747	AAFD	23	.	INX H	
8748	AAFE	22	CB FF	SHLD ZTOPLN ;SET NEW TOPLINE	
8749	AB01	3A	02 FB	LDA MUFLD ;PUT CURSOR IN CURRENT FIELD	
8750	AB04	C3	ED AD	JMP PUTFLD	
8751	AB07	C9	.	RET	
8752	AB08	.	.	*****	
8753	AB08	.	.	; MUOFF--TURN AUTO PLOT MENU OFF	
8754	AB08	.	.	*****	
8755	AB08	.	.	MUOFF EQU \$	
8756	AB08	CD	C7 AE	CALL CLOSE ;CLOSE CURRENT FIELD	
8757	AB0B	3E	01 .	MVI A,MENUON ;CLEAR MENU FLAG	
8758	AB0D	CD	C2 B8	CALL CLAPF2	
8759	AB10	2A	00 FB	LHLD TOPSAV ;RESTORE TOP LINE	
8760	AB13	22	CB FF	SHLD ZTOPLN	
8761	AB16	2A	06 FB	LHLD CURSAV ;RESTORE CURSOR	

13255
2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

```
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
8762	AB19	22	C0	FF	SHLD ZCUROW	238
8763	AB1C	C3	43	00	JMP ZRSTDP ;RESTURE NORMAL DISPLAY	

```
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 239
=====
8765     AB1F      .      .      .      ;*****
8766     AB1F      .      .      .      ; MUTODM--TRANSFER MENU FROM ROM TO DISPLAY MEM
8767     AB1F      .      .      .      ;*****
8768     AB1F      .      .      .      MUTODM EQU $
8769     AB1F      21     3C     AB      LXI H,APMENU ;SOURCE OF MENU
8770     AB22      11     DA     FA      LXI D,MUBUF  ;DESTINATION IN DISPLAY MEM
8771     AB25      01     4D     02      LXI B,MULEN  ;LENGTH
8772     AB28      .      .      .      ;*****
8773     AB28      .      .      .      ; XFER--TRANSFER FROM ONE AREA TO ANOTHER
8774     AB28      .      .      .      ; IN REVERSE ORDER
8775     AB28      .      .      .      ; ENTRY HL = SOURCE POINTER
8776     AB28      .      .      .      ; DE = DESTINATION POINTER
8777     AB28      .      .      .      ; BC = LENGTH
8778     AB28      .      .      .      ;*****
8779     AB28      .      .      .      XFER EQU $
8780     AB28      7E     .      .      MOV A,M      ;FETCH BYTE
8781     AB29      23     .      .      INX H        ;UPDATE SOURCE POINTER
8782     AB2A      12     .      .      STAX D       ;STORE IN NEW AREA
8783     AB2B      1B     .      .      DCX D        ;UPDATE DESTINATION POINTER
8784     AB2C      0B     .      .      DCX B        ;UPDATE COUNT
8785     AB2D      78     .      .      MOV A,B      ;SEE IF DONE
8786     AB2E      B1     .      .      ORA C
8787     AB2F      C2     28     AB      JNZ XFER     ;NOT DONE YET
8788     AB32      C9     .      .      RET
8789     AB33      .      .      .      ;*****
8790     AB33      .      .      .      ; XFER2-- TRANSER FROM DISPLAY MEMORY TO BUFFER
8791     AB33      .      .      .      ; IN REVERSE ORDER UNTIL EOL FOUND
8792     AB33      .      .      .      ; ENTRY HL = SOURCE POINTER
8793     AB33      .      .      .      ; DE = DESTINATION POINTER
8794     AB33      .      .      .      ;*****
8795     AB33      .      .      .      XFER2 EQU $
8796     AB33      7E     .      .      MOV A,M      ;FETCH CHAR
8797     AB34      12     .      .      STAX D       ;STORE IT
8798     AB35      B7     .      .      ORA A        ;IS IT EOL?
8799     AB36      F8     .      .      RM           ;YES
8800     AB37      2B     .      .      DCX H        ;NO, UPDATE SOURCE POINTER
8801     AB38      13     .      .      INX D        ;UPDATE DESTINATION POINTER
8802     AB39      C3     33     AB      JMP XFER2    ;DO NEXT BYTE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 240
8804	AB3C	.	.	.	APMENU EQU \$	
8805	AB3C	20	20	20	DB ' ,IVON	
8806	AB43	41	55	54	DB 'AUTO PLOT',EOL,EOL	
8807	AB4E	41	2E	20	DB 'A. PLOT SPECIFICATION',EOL	
8808	AB64	20	31	2E	DB ' 1. NO. OF COLUMNS '	
8809	AB78	82	.	.	DB IVON	
8810	AB79	.	.	.	XFLD1 EQU \$	
8811	AB79	20	20	20	DB ' ,EOL	
8812	AB7F	20	32	2E	DB ' 2. X IS COLUMN '	
8813	AB93	82	.	.	DB IVON	
8814	AB94	.	.	.	XFLD2 EQU \$	
8815	AB94	20	20	20	DB ' ,EOL	
8816	AB9A	20	33	2E	DB ' 3. Y IS COLUMN '	
8817	ABAE	82	.	.	DB IVON	
8818	ABAF	.	.	.	XFLD3 EQU \$	
8819	ABAF	20	20	20	DB ' ,EOL	
8820	ABB5	20	34	2E	DB ' 4. LINE TYPE (1-9) '	
8821	ABC9	82	.	.	DB IVON	
8822	ABCA	.	.	.	XFLD4 EQU \$	
8823	ABCA	20	20	20	DB ' ,EOL	
8824	ABD0	20	35	2E	DB ' 5. MIN X '	
8825	ABDA	82	.	.	DB IVON	
8826	ABDB	.	.	.	XFLD5 EQU \$	
8827	ABDB	20	20	20	DB ' ,EOL	
8828	ABEB	20	36	2E	DB ' 6. MAX X '	
8829	ABF5	82	.	.	DB IVON	
8830	ABF6	.	.	.	XFLD6 EQU \$	
8831	ABF6	20	20	20	DB ' ,EOL	
8832	AC06	20	37	2E	DB ' 7. MIN Y '	
8833	AC10	82	.	.	DB IVON	
8834	AC11	.	.	.	XFLD7 EQU \$	
8835	AC11	20	20	20	DB ' ,EOL	
8836	AC21	20	38	2E	DB ' 8. MAX Y '	
8837	AC2B	82	.	.	DB IVON	
8838	AC2C	.	.	.	XFLD8 EQU \$	
8839	AC2C	20	20	20	DB ' ,EOL,EOL	
8840	AC3D	42	2E	20	DB 'B. AXES SPECIFICATION',EOL	
8841	AC53	20	31	2E	DB ' 1. UNITS BETWEEN X LABELS '	
8842	AC6E	82	.	.	DB IVON	
8843	AC6F	.	.	.	XFLD9 EQU \$	
8844	AC6F	20	20	20	DB ' ,EOL	
8845	AC7F	20	32	2E	DB ' 2. UNITS BETWEEN X TICS '	
8846	AC9A	82	.	.	DB IVON	
8847	AC9B	.	.	.	XFLD10 EQU \$	
8848	AC9B	20	20	20	DB ' ,EOL	
8849	ACAB	20	33	2E	DB ' 3. UNITS BETWEEN Y LABELS '	
8850	ACC6	82	.	.	DB IVON	
8851	ACC7	.	.	.	XFLD11 EQU \$	
8852	ACC7	20	20	20	DB ' ,EOL	
8853	ACD7	20	34	2E	DB ' 4. UNITS BETWEEN Y TICS '	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 241
=====
8854      ACF2      82      .      .      DB      IVON
8855      ACF3      .      .      .      XFLD12 EQU $
8856      ACF3      20      20      20      DB      '                                     ',EOL,EOL
8857      AD04      43      2E      20      DB      'C. PLOT OPTIONS ',EOL
8858      AD15      20      31      2E      DB      ' 1. SKIP FIRST '
8859      AD25      82      .      .      DB      IVON
8860      AD26      .      .      .      XFLD13 EQU $
8861      AD26      20      20      20      DB      '          ',IVOFF
8862      AD2C      20      4C      49      DB      ' LINES OF TEXT',EOL
8863      AD3B      20      32      2E      DB      ' 2. STOP AFTER '
8864      AD4B      82      .      .      DB      IVON
8865      AD4C      .      .      .      XFLD14 EQU $
8866      AD4C      20      20      20      DB      '          ',IVOFF
8867      AD52      20      50      4F      DB      ' POINTS',EOL
8868      AD5A      20      33      2E      DB      ' 3. DRAW GRID? '
8869      AD6A      82      .      .      DB      IVON
8870      AD6B      .      .      .      XFLD15 EQU $
8871      AD6B      20      20      20      DB      '          ',EOL
8872      AD71      20      34      2E      DB      ' 4. FROM DSPLY? '
8873      AD81      82      .      .      DB      IVON
8874      AD82      .      .      .      XFLD16 EQU $
8875      AD82      20      20      20      DB      '          ',EOL
8876      AD88      CE      .      .      DB      ZEOP
8877      024D      .      .      .      MULEN EQU $-APMENU ;LENGTH OF THIS MENU
8878      F88D      .      .      .      NEWLIM EQU MUBUF-MULEN ;NEW TOP OF DISPLAY MEM
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 242
8880	AD89	.	.	. ;*****	
8881	AD89	.	.	. ; MUESC--ESCAPE RECEIVED IN MENU ROUTINE	
8882	AD89	.	.	. ; ONLY CURSOR POSITIONING SEQ ARE ALLOWED	
8883	AD89	.	.	. ; ESC A, B, B, C, D, OR H	
8884	AD89	.	.	. ; IF ESC * OR ESC & IS RECEIVED, IT WILL BE	
8885	AD89	.	.	. ; EXECUTED.	
8886	AD89	.	.	. ;*****	
8887	AD89	.	.	. MUESC EQU \$	
8888	AD89	21	D1	FF LXI H,ZESCFG ;SET 2 CHAR ESC SEQ COUNTER	
8889	AD8C	36	02	. MVI M,2 ;GOING	
8890	AD8E	21	30	62 LXI H,METB ;USE MENU ESCAPE TABLE	
8891	AD91	22	D2	FF SHLD ZRNGTA	
8892	AD94	C9	.	. RET	
8893	AD95	.	.	. ;*****	
8894	AD95	.	.	. ; APGSEQ--ESC * RECEIVED WHILE AUTOPLLOT	
8895	AD95	.	.	. ; MENU UP. ONLY SEQUENCE ALLOW IS ESC * A	
8896	AD95	.	.	. ; TO LOAD MENU. ABORT MENU AND SEQUENCE ON ANY	
8897	AD95	.	.	. ; OTHER	
8898	AD95	.	.	. ;*****	
8899	AD95	.	.	. APGSEQ EQU \$	
8900	AD95	21	60	62 LXI H,APGTAB ;LOAD RANGE TABLE	
8901	AD98	C3	86	63 JMP SETRTB	
8902	AD9B	.	.	. ;*****	
8903	AD9B	.	.	. ; HOME--PUT CURSOR IN FIELD 1, COLUMN 1	
8904	AD9B	.	.	. ; ENTRY--DONT CARE	
8905	AD9B	.	.	. ; EXIT---ALL REGISTERS DESTROYED	
8906	AD9B	.	.	. ;*****	
8907	AD9B	.	.	. HOME EQU \$	
8908	AD9B	CD	C7	AE CALL CLOSE ;NO--CLOSE THE CURRENT FIELD	
8909	AD9E	3E	00	. MVI A,FLD1 ;PUT CURSOR IN FIRST FIELD	
8910	ADA0	C3	ED	AD JMP PUTFLD ;FIRST COLUMN	
8911	ADA3	.	.	. ;*****	
8912	ADA3	.	.	. ; HOMEDN--MOVE CURSOR TO LAST FIELD	
8913	ADA3	.	.	. ;*****	
8914	ADA3	.	.	. HOMEDN EQU \$	
8915	ADA3	CD	C7	AE CALL CLOSE ;CLOSE CURRENT FIELD	
8916	ADA6	3E	0F	. MVI A,BOTFLD ;PUT CURSOR IN LAST FIELD	
8917	ADA8	C3	ED	AD JMP PUTFLD	
8918	ADAB	.	.	. ;*****	
8919	ADAB	.	.	. ; CLRFLD--CLEAR CURRENT FIELD FROM CURSOR	
8920	ADAB	.	.	. ;*****	
8921	ADAB	.	.	. CLRFLD EQU \$	
8922	ADAB	3A	C1	FF LDA ZCURCL ;SAVE CURRENT CURSOR COL	
8923	ADAE	F5	.	. PUSH PSW	
8924	ADAF	.	.	. CLR010 EQU \$	
8925	ADAF	0E	20	. MVI C,40Q ;LOAD A SPACE INTO THE	
8926	ADB1	CD	A8	AE CALL ADDCH1 ;FIELD	
8927	ADB4	F2	AF	AD JP CLR010 ;LOOP TILL END OF FIELD	
8928	ADB7	F1	.	. POP PSW ;RESTORE CURSOR	
8929	ADB8	C3	89	AE JMP PUTCOL ;COLUMN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 243
=====
8931     ADBB      . . .      ;*****
8932     ADBB      . . .      ; MOVST--MOVE CURSOR TO START (COL1) OF CURRENT
8933     ADBB      . . .      ; FIELD
8934     ADBB      . . .      ;*****
8935     ADBB      . . .      MOVST EQU $
8936     ADBB      3A 03 FB      LDA COL1          ;FETCH STARTING COLUMN
8937     ADBE      . . .      MOVST1 EQU $
8938     ADBE      32 C1 FF      STA ZCURCL        ;UPDATE COLUMN
8939     ADC1      32 00 87      STA ZIOCCL        ;SEND TO HW
8940     ADC4      C9 . .      RET
8941     ADC5      . . .      ;*****
8942     ADC5      . . .      ; MOVDN--MOVE CURSOR DOWN ONE FIELD
8943     ADC5      . . .      ;*****
8944     ADC5      . . .      MOVDN EQU $
8945     ADC5      3A 02 FB      LDA MUFLD         ;FETCH CURRENT FIELD
8946     ADC8      FE 0F .      CPI BOTFLD        ;ALREADY IN LAST FIELD?
8947     ADCA      C8 . .      RZ                ;YES--IGNORE COMMAND
8948     ADCB      CD C7 AE      CALL CLOSE        ;NO--CLOSE CURRENT FIELD
8949     ADCE      3C . .      INR A             ;INCREMENT FIELD
8950     ADCF      C3 ED AD      JMP PUTFLD        ;PUT CURSOR IN NEW FIELD
8951     ADD2      . . .      ;*****
8952     ADD2      . . .      ; MOVUP--MOVE CURSOR UP ONE FIELD
8953     ADD2      . . .      ;*****
8954     ADD2      . . .      MOVUP EQU $
8955     ADD2      3A 02 FB      LDA MUFLD         ;FETCH CURRENT FIELD
8956     ADD5      FE 00 .      CPI FLD1          ;ALREADY IN FIRST FIELD?
8957     ADD7      C8 . .      RZ                ;YES--IGNORE COMMAND
8958     ADD8      CD C7 AE      CALL CLOSE        ;NO--CLOSE CURRENT FIELD
8959     ADDB      3D . .      DCR A             ;UP ONE FIELD
8960     ADDC      C3 ED AD      JMP PUTFLD        ;PUT CURSOR IN NEW FIELD
8961     ADDF      . . .      ;*****
8962     ADDF      . . .      ; MOVRT--MOVE RIGHT ONE COL
8963     ADDF      . . .      ;*****
8964     ADDF      . . .      MOVRT EQU $
8965     ADDF      3A C1 FF      LDA ZCURCL        ;FETCH CURRENT COLUMN
8966     ADE2      3C . .      INR A             ;RIGHT ONE
8967     ADE3      C3 89 AE      JMP PUTCOL        ;UPDATE CURSOR COL
8968     ADE6      . . .      ;*****
8969     ADE6      . . .      ; MOVLFT--MOVE CURSOR LEFT ON COLUMN
8970     ADE6      . . .      ;*****
8971     ADE6      . . .      MOVLFT EQU $
8972     ADE6      3A C1 FF      LDA ZCURCL        ;FETCH CURRENT COLUMN
8973     ADE9      3D . .      DCR A             ;LEFT ONE
8974     ADEA      C3 89 AE      JMP PUTCOL        ;UPDATE COLUMN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
8976	ADED	.	.	*****
8977	ADED	.	.	; PUTFLD--PUT CURSOR IN MENU FIELD
8978	ADED	.	.	; MOVE CURSOR TO COLUMN 1 OF FIELD
8979	ADED	.	.	; ENTRY A = FIELD
8980	ADED	.	.	*****
8981	ADED	.	.	PUTFLD EQU \$
8982	ADED	21	02	FB LXI H,MUFLD ;STORE NEW FIELD
8983	ADF0	77	.	MOV M,A
8984	ADF1	.	.	;COMPUTE PHYSICAL CURSOR ROW FROM FIELD NUMBER
8985	ADF1	5F	.	MOV E,A
8986	ADF2	16	00	. MVI D,0 ;INDEX TO ROW TABLE
8987	ADF4	.	.	; IF MENU IS NOT ON, DONT CHANGE CURSOR POSITION
8988	ADF4	CD	EE	B8 CALL MUCHK ;IS IT ON?
8989	ADF7	CA	10	AE JZ PTF010 ;NO
8990	ADFA	21	39	AE LXI H,FLOTB ;BASE OF TABLE
8991	ADFD	19	.	DAD D
8992	ADFE	7E	.	MOV A,M ;FETCH ACTUAL ROW
8993	ADFF	32	C0	FF STA ZCURROW ;STORE NEW ROW
8994	AE02	32	20	87 STA ZIOCRW ;SEND TO HW
8995	AE05	.	.	; FETCH FIRST COLUMN FOR THIS FIELD FROM TABLE
8996	AE05	21	29	AE LXI H,COLTB ;BASE OF COLUMN TABLE
8997	AE08	19	.	DAD D ;HL = POINTER TO FIRST COL
8998	AE09	7E	.	MOV A,M ;FETCH IT
8999	AE0A	32	03	FB STA COL1
9000	AE0D	CD	BE	AD CALL MOVST1 ;MOVE TO START OF FIELD
9001	AE10	.	.	; USING FIELD NUMBER, COMPUTE ACTUAL ADDRESS IN
9002	AE10	.	.	; DISPLAY MEMORY OF ASCII FIELD, AND LOCATION
9003	AE10	.	.	; OF PARAMETER BUFFER POINTER
9004	AE10	.	.	PTF010 EQU \$
9005	AE10	EB	.	XCHG ;HL = INDEX
9006	AE11	29	.	DAD H ;INDEX * 2
9007	AE12	29	.	DAD H ;INDEX * 4
9008	AE13	EB	.	XCHG ;DE = INDEX
9009	AE14	21	49	AE LXI H,ADDRTB ;BASE OF TABLE
9010	AE17	19	.	DAD D ;+ 2 * INDEX
9011	AE18	5E	.	MOV E,M ;FETCH DISPLAY POINTER
9012	AE19	23	.	INX H
9013	AE1A	56	.	MOV D,M
9014	AE1B	EB	.	XCHG ;HL=PTR TO START OF FIELD
9015	AE1C	22	FC	FA SHLD DSPFLD ;STORE DISPLAY POINTER
9016	AE1F	EB	.	XCHG ;HL = TABLE POINTER
9017	AE20	23	.	INX H
9018	AE21	5E	.	MOV E,M
9019	AE22	23	.	INX H
9020	AE23	56	.	MOV D,M
9021	AE24	EB	.	XCHG ;HL = POINTER TO BUFFER
9022	AE25	22	FE	FA SHLD APBUF ;STORE PARAM BUFFER PTR
9023	AE28	C9	.	RET


```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 245
=====
9025      AE29      . . .      ;*****
9026      AE29      . . .      ; COLTB--PHYSICAL FIRST COLUMN ON SCREEN FOR
9027      AE29      . . .      ; EACH FIELD
9028      AE29      . . .      ;*****
9029      AE29      . . .      COLTB EQU $
9030      AE29      14 14 14      DB 20,20,20,20
9031      AE2D      0A 0A 0A      DB 10,10,10,10
9032      AE31      1B 1B 1B      DB 27,27,27,27
9033      AE35      10 10 10      DB 16,16,16,16
9034      AE39      . . .      ;*****
9035      AE39      . . .      ; FLDTB--PHYSICAL ROW ON SCREEN FOR EACH FIELD
9036      AE39      . . .      ;*****
9037      AE39      . . .      FLDTB EQU $
9038      AE39      03 04 05      DB 3,4,5,6
9039      AE3D      07 08 09      DB 7,8,9,10
9040      AE41      0D 0E 0F      DB 13,14,15,16
9041      AE45      13 14 15      DB 19,20,21,22
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 246
9043	AE49	.	.	*****	
9044	AE49	.	.	; ADDRTB-FIRST ENTRY IS ACTUAL ADDRESS IN	
9045	AE49	.	.	; DISPLAY MEMORY OF START OF FIELD	
9046	AE49	.	.	; SECOND IS POINTER OF PARAMETER BUFFER FOR	
9047	AE49	.	.	; THIS FIELD	
9048	AE49	.	.	*****	
9049	AE49	.	.	ADDRTB EQU \$	
9050	AE49	9D	FA	DW MUBUF-XFLD1+APMENU	
9051	AE48	C3	FB	DW APB1	
9052	AE4D	82	FA	DW MUBUF-XFLD2+APMENU	
9053	AE4F	C2	FB	DW APB2	
9054	AE51	67	FA	DW MUBUF-XFLD3+APMENU	
9055	AE53	C1	FB	DW APB3	
9056	AE55	4C	FA	DW MUBUF-XFLD4+APMENU	
9057	AE57	C0	FB	DW APB4	
9058	AE59	3B	FA	DW MUBUF-XFLD5+APMENU	
9059	AE5B	BC	FB	DW APB5	
9060	AE5D	20	FA	DW MUBUF-XFLD6+APMENU	
9061	AE5F	B8	FB	DW APB6	
9062	AE61	05	FA	DW MUBUF-XFLD7+APMENU	
9063	AE63	B4	FB	DW APB7	
9064	AE65	EA	F9	DW MUBUF-XFLD8+APMENU	
9065	AE67	B0	FB	DW APB8	
9066	AE69	A7	F9	DW MUBUF-XFLD9+APMENU	
9067	AE6B	AC	FB	DW APB9	
9068	AE6D	7B	F9	DW MUBUF-XFLD10+APMENU	
9069	AE6F	A8	FB	DW APB10	
9070	AE71	4F	F9	DW MUBUF-XFLD11+APMENU	
9071	AE73	A4	FB	DW APB11	
9072	AE75	23	F9	DW MUBUF-XFLD12+APMENU	
9073	AE77	A0	FB	DW APB12	
9074	AE79	F0	F8	DW MUBUF-XFLD13+APMENU	
9075	AE7B	9E	FB	DW APB13	
9076	AE7D	CA	F8	DW MUBUF-XFLD14+APMENU	
9077	AE7F	9C	FB	DW APB14	
9078	AE81	AB	F8	DW MUBUF-XFLD15+APMENU	
9079	AE83	9A	FB	DW APB15	
9080	AE85	94	F8	DW MUBUF-XFLD16+APMENU	
9081	AE87	98	FB	DW APB16	
9082	AE89	.	.	;	
9083	F9A7	.	.	XLBASC EQU MUBUF-XFLD9+APMENU ;X LABEL IN MENU	
9084	F94F	.	.	YLBASC EQU MUBUF-XFLD11+APMENU ;Y LABEL IN MENU	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 247
9086	AE89	.	.	*****	
9087	AE89	.	.	; PUTCOL--PUT CURSOR IN COLUMN ON MENU	
9088	AE89	.	.	; DO BOUNDS CHECK TO INSURE IT REMAINS IN FIELD	
9089	AE89	.	.	; ENTRY A = COL	
9090	AE89	.	.	; EXIT M => AT END OF FIELD	
9091	AE89	.	.	*****	
9092	AE89	.	.	PUTCOL EQU \$	
9093	AE89	.	.	; TEST FOR LEFT OF FIRST COL	
9094	AE89	21	03 FB	LXI H,COL1	
9095	AE8C	BE	.	CMP M ;TOO SMALL?	
9096	AE8D	D8	.	RC ;YES, LEAVE COL AS IS	
9097	AE8E	.	.	; TEST FOR RIGHT OF EOL = END OF FIELD	
9098	AE8E	.	.	; SEE IF NEW COL IS AT EOL	
9099	AE8E	4F	.	MOV C,A ;SAVE NEW COL	
9100	AE8F	96	.	SUB M ;CONVERT TO RELATIVE POSITIO	
9101	AE90	2F	.	CMA ;WANT -VALUE	
9102	AE91	5F	.	MOV E,A	
9103	AE92	16	FF	MVI D,377Q	
9104	AE94	13	.	INX D ;DE = OFFSET FROM ST OF FLD	
9105	AE95	2A	FC FA	LHLD DSPFLD ;ST OF FIELD	
9106	AE98	19	.	DAD D ;HL = POINTER TO DSP MEM	
9107	AE99	7E	.	MOV A,M ;SEE IF THERES AN EOL THERE	
9108	AE9A	B7	.	ORA A ;EOL OR IVOFF?	
9109	AE9B	F8	.	RM ;YES, END OF FIELD	
9110	AE9C	.	.	;COL IN BOUNDS, STORE IT	
9111	AE9C	79	.	MOV A,C ;RECALL NEW COLUMN	
9112	AE9D	32	C1 FF	STA ZCURCL ;STORE CURRENT COLUMN	
9113	AEA0	32	00 87	STA ZIOCCL ;SEND TO HW	
9114	AEA3	C9	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 248
9116	AEA4	.	.	*****	
9117	AEA4	.	.	; ADDCHR-- ADD CHARACTER TO MENU AT CURRENT	
9118	AEA4	.	.	; CURSOR POSITION, ADVANCE CURSOR	
9119	AEA4	.	.	*****	
9120	AEA4	.	.	ADDCHR EQU \$	
9121	AEA4	21	88	FF LXI H,ZCHAR ;GET THE CHARACTER	
9122	AEA7	4E	.	MOV C,M ;SAVE IN C REG	
9123	AEA8	.	.	ADDCH1 EQU \$	
9124	AEA8	.	.	;CHECK SOURCE OF CHAR. IF NOT FROM KEYBOARD, TURN	
9125	AEA8	.	.	;THE MENU OFF	
9126	AEA8	CD	C3	00 CALL ZDCIO ;FROM KEYBOARD?	
9127	AEA8	C2	1D	B9 JNZ CHKCH1 ;NO, RESTORE NORMAL DISPLAY	
9128	AEA8	.	.	; COMPUTE DESTINATION ADDRESS IN DISPALY MEM	
9129	AEA8	21	03	FB LXI H,COL1 ;FIRST COLUMN	
9130	AEB1	3A	C1	FF LDA ZCURCL ;ABSOLUTE COLUMN NUMBER	
9131	AEB4	96	.	SUB M ;CONVERT TO RELATIVE LOCATIO	
9132	AEB5	2A	FC	FA LHLD DSPFLD ;START OF FIELD	
9133	AEB8	2F	.	CMA ;MENT WITHIN FIELD	
9134	AEB9	5F	.	MOV E,A ;WANT -DISPALCEMENT	
9135	AEBA	16	FF	MVI D,377Q	
9136	AEB3	13	.	INX D	
9137	AEBD	19	.	DAD D ;HL = POINTER TO DEST.	
9138	AEBE	71	.	MOV M,C ;STORE THE CHAR	
9139	AEBF	.	.	;ADVANCE CURSOR	
9140	AEBF	C3	DF	AD JMP MOVRT ;MOVE RIGHT ONE	
9141	AEC2	.	.	*****	
9142	AEC2	.	.	; ADDE--LOWER CASE E RECEIVED. REPLACE WITH CAP	
9143	AEC2	.	.	*****	
9144	AEC2	.	.	ADDE EQU \$	
9145	AEC2	0E	45	MVI C,105Q ;FAKE A CAP E	
9146	AEC4	C3	A8	AE JMP ADDCH1 ;PROCESS AS USUAL	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	
9148	AEC7	.	.	.	;*****
9149	AEC7	.	.	.	; CLOSE--CLOSE CURRENT FIELD
9150	AEC7	.	.	.	; CONVERT ASCII TO INTEGER OR FP
9151	AEC7	.	.	.	;*****
9152	AEC7	.	.	.	CLOSE EQU \$
9153	AEC7	F5	.	.	PUSH PSW ;SAVE A REG
9154	AEC8	2A	FC	FA	LHLD DSPFLD ;POINTER TO CURRENT FIELD
9155	AECB	CD	DF	AE	CALL TRIM ;DELETE LEADING BLANKS
9156	AECE	CD	E8	AE	CALL FPCHK ;SEE IF THIS IS A FP FIELD
9157	AED1	CA	DA	AE	JZ CLS010 ;YES--DO FP CONVERSION
9158	AED4	CD	15	AF	CALL ICNVRT ;NO--DO INTEGER CONVERSION
9159	AED7	C3	DD	AE	JMP CLS020
9160	AEDA	.	.	.	CLS010 EQU \$
9161	AEDA	CD	03	AF	CALL FPCNVT ;DO FLOATING PT CONVERSION
9162	AEDD	.	.	.	CLS020 EQU \$
9163	AEDD	F1	.	.	POP PSW ;RECALL A REG
9164	AEDE	C9	.	.	RET
9165	AEDF	.	.	.	;*****
9166	AEDF	.	.	.	; TRIM--DELETE LEADING BLANKS IN ASCII FIELD
9167	AEDF	.	.	.	; ENTRY HL = POINTER TO FIRST CHAR IN FIELD
9168	AEDF	.	.	.	; EXIT HL = POINTER TO FIRST NON BLANK CHAR
9169	AEDF	.	.	.	;*****
9170	AEDF	.	.	.	TRIM EQU \$
9171	AEDF	23	.	.	INX H ;FUDGE FOR FIRST TIME THRU
9172	AEEO	.	.	.	TRM010 EQU \$
9173	AEEO	2B	.	.	DCX H ;ADVANCE TO NEXT FIELD
9174	AEEO	7E	.	.	MOV A,M ;FETCH CHAR
9175	AEEO	FE	20	.	CPI 40Q ;SPACE?
9176	AEEO	CA	E0	AE	JZ TRM010 ;YES, IGNORE
9177	AEEO	C9	.	.	RET
9178	AEEO	.	.	.	;*****
9179	AEEO	.	.	.	; FPCHK--SEE IF CURRENT FIELD IS FLOATING POINT
9180	AEEO	.	.	.	; EXIT Z => FLOATING POINT
9181	AEEO	.	.	.	;*****
9182	AEEO	.	.	.	FPCHK EQU \$
9183	AEEO	3A	02	FB	LDA MUFLD ;FETCH CURRENT FIELD
9184	AEEO	FE	04	.	CPI XMINFD
9185	AEEO	C8	.	.	RZ
9186	AEEO	FE	05	.	CPI XMAXFD
9187	AEEO	C8	.	.	RZ
9188	AEEO	FE	06	.	CPI YMINFD
9189	AEEO	C8	.	.	RZ
9190	AEEO	FE	07	.	CPI YMAXFD
9191	AEEO	C8	.	.	RZ
9192	AEEO	FE	09	.	CPI XTICFD
9193	AEEO	C8	.	.	RZ
9194	AEEO	FE	0B	.	CPI YTICFD
9195	AEEO	C8	.	.	RZ
9196	AEEO	FE	08	.	CPI XLBLFD
9197	AEEO	C8	.	.	RZ

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
9198     AF00     FE 0A .          CPI YLBLFD
9199     AF02     C9 . .          RET
=====
```

PAGE 250

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 251
9201	AF03	.	.	.	;*****
9202	AF03	.	.	.	; FPCNVT--CONVERT ASCII FIELD TO FLOATING POINT
9203	AF03	.	.	.	; ENTRY HL = POINTER TO FIRST CHAR IN FIELD
9204	AF03	.	.	.	;*****
9205	AF03	.	.	.	FPCNVT EQU \$
9206	AF03	.	.	.	; DATA IS STORED BACKWARDS IN DISPLAY MEMORY
9207	AF03	.	.	.	; DATA FOR INP CANNOT CROSS PAGE BOUNDRIES
9208	AF03	.	.	.	; SO, XFER FROM MENU TO ONE PAGE BUFFER
9209	AF03	.	.	.	; (AND PUT INTO PROPER ORDER)
9210	AF03	11	49	FB	LXI D,NUMBUF+60 ;NEW SINGLE PAGE BUFFER
9211	AF06	CD	33	AB	CALL XFER2 ;GO FROM DISPLAY TO BUFFER
9212	AF09	21	49	FB	LXI H,NUMBUF+60 ;NEW DATA BUFFER
9213	AF0C	CD	3F	B1	CALL FPINP ;CALL CONVERSION ROUTINE
9214	AF0F	.	.	.	; TRANSFER FROM SCATCH BUFFER TO PROPER
9215	AF0F	.	.	.	; FLOATING POINT BUFFER
9216	AF0F	2A	FE	FA	LHLD APBUF ;BUFFER FOR THIS FIELD
9217	AF12	C3	3E	BD	JMP STR ;STORE FLOATING POINT VALUE
9218	AF15	.	.	.	;*****
9219	AF15	.	.	.	; ICNVRT--CONVERT INTEGER ASCII FIELD TO BIN
9220	AF15	.	.	.	; LEADING +,- SPACE IGNORED
9221	AF15	.	.	.	; ANY OTHER NON 0-9 TERMINATES
9222	AF15	.	.	.	; AFTER FIRST DIGIT, E OR . OR SIGN TERMINATES
9223	AF15	.	.	.	; ENTRY HL = POINTER TO START OF STRING
9224	AF15	.	.	.	;*****
9225	AF15	.	.	.	ICNVRT EQU \$
9226	AF15	11	00	00	LXI D,0 ;INITIALOZE RESULT TO 0
9227	AF18	23	.	.	INX H ;FUDGE FOR 1ST TIME THRU
9228	AF19	0E	00	.	MVI C,0 ;C = NUMBER IN PROGRESS FLAG
9229	AF18	.	.	.	ICN010 EQU \$
9230	AF18	2B	.	.	DCX H ;ADVANCE TO NEXT FIELD
9231	AF1C	7E	.	.	MOV A,M ;FETCH CHAR
9232	AF1D	FE	20	.	CPI 40Q ;SPACE?
9233	AF1F	CA	3E	AF	JZ ICN020 ;SEE IF NUMBER HAS STARTED
9234	AF22	FE	2B	.	CPI 53Q ;+ ?
9235	AF24	CA	3E	AF	JZ ICN020 ;SEE IF NUMBER HAS STARTED
9236	AF27	FE	2D	.	CPI 55Q ;- ?
9237	AF29	CA	3E	AF	JZ ICN020 ;SEE IF NUMBER HAS STARTED
9238	AF2C	FE	30	.	CPI 60Q ;DIGIT?
9239	AF2E	DA	43	AF	JC ICN030 ;NO TERMINATE
9240	AF31	FE	3A	.	CPI 72Q ;DIGIT?
9241	AF33	D2	43	AF	JNC ICN030 ;NO--TERMINATE
9242	AF36	.	.	.	;HAVE DIGIT, ADD TO ACCUMULATED BINARY
9243	AF36	CD	4E	AF	CALL BCDBIN ;ADD TO RUNNING SUM
9244	AF39	0E	FF	.	MVI C,377Q ;SET NUMBER IN PROGRESS FLG
9245	AF3B	C3	1B	AF	JMP ICN010 ;GET NEXT NUM
9246	AF3E	.	.	.	ICN020 EQU \$
9247	AF3E	.	.	.	;SPACE, + OR - RECEIVED. IF NUMBER HASNT
9248	AF3E	.	.	.	; STARTED YET, IGNORE, OTHERWISE, TERMINATE
9249	AF3E	79	.	.	MOV A,C ;TEST NUMBER IN PROGRESS FLA
9250	AF3F	B7	.	.	ORA A ;HAS NUMBER STARTED YET?

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 252
9251	AF40	CA	1B	AF	JZ ICN010 ;NO--IGNORE, GET NEXT CHAR
9252	AF43	.	.	.	ICN030 EQU \$
9253	AF43	.	.	.	;TERMINATE COINVERSION
9254	AF43	.	.	.	; DE = BINARY VALUE
9255	AF43	2A	FE	FA	LHLD APBUF ;HL = POINTER TO PROPER BUF
9256	AF46	73	.	.	MOV M,E ;STORE LSBYTE
9257	AF47	.	.	.	; SEE IF THIS FIELD REQUIRES 2 BYTES OF STORAGE
9258	AF47	CD	5F	AF	CALL INTCHK
9259	AF4A	C0	.	.	RNZ ;NO, DONT STORE ANY MORE
9260	AF4B	23	.	.	INX H ;YES, STORE MSBYTE
9261	AF4C	72	.	.	MOV M,D
9262	AF4D	C9	.	.	RET
9263	AF4E	.	.	.	;*****
9264	AF4E	.	.	.	; BCDBIN--CONVERT BCD TO BINARY
9265	AF4E	.	.	.	; ENTRY A = NEXT DIGIT
9266	AF4E	.	.	.	; DE = RUNNING SUMM (INITIALIZED TO 0)
9267	AF4E	.	.	.	;*****
9268	AF4E	.	.	.	BCDBIN EQU \$
9269	AF4E	E5	.	.	PUSH H ;SAVE H REG
9270	AF4F	EB	.	.	XCHG ;HL = RUNNING SUM
9271	AF50	.	.	.	;MULTIPLY CURRENT VALUE BY 10
9272	AF50	E5	.	.	PUSH H ;SAVE H
9273	AF51	29	.	.	DAD H ;H * 2
9274	AF52	29	.	.	DAD H ;H * 4
9275	AF53	D1	.	.	POP D
9276	AF54	19	.	.	DAD D ;H * 5
9277	AF55	29	.	.	DAD H ;H * 10
9278	AF56	.	.	.	;NOW ADD IN NEW CHAR
9279	AF56	E6	0F	.	ANI 170 ;DELETE LEADING BITS
9280	AF58	5F	.	.	MOV E,A ;DE = NEW CHAR
9281	AF59	16	00	.	MVI D,0
9282	AF5B	19	.	.	DAD D ;ADD TO CURRENT
9283	AF5C	EB	.	.	XCHG ;DE = NEW BINARY VALUE
9284	AF5D	E1	.	.	POP H ;RESTORE H
9285	AF5E	C9	.	.	RET
9286	AF5F	.	.	.	;*****
9287	AF5F	.	.	.	; INTCHK--SEE IF INTEGER FIELD IS STORED AS
9288	AF5F	.	.	.	; 16 BIT VALUE
9289	AF5F	.	.	.	; EXIT Z => STORE 16 BITS
9290	AF5F	.	.	.	;*****
9291	AF5F	.	.	.	INTCHK EQU \$
9292	AF5F	3A	02	FB	LDA MUFLO ;FETCH CURRENT FIELD
9293	AF62	FE	0D	.	CPI CNTFLD ;POINT COUNT??
9294	AF64	C8	.	.	RZ
9295	AF65	FE	0C	.	CPI SKPFLD ;LINES TO SKIP??
9296	AF67	C8	.	.	RZ
9297	AF68	FE	0F	.	CPI FRMFLD ;FROM DISPLAY?
9298	AF6A	C8	.	.	RZ
9299	AF6B	FE	0E	.	CPI GRIDFD ;WANT GRID?
9300	AF6D	C9	.	.	RET

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 253
9302	AF6E	.	.	*****	
9303	AF6E	.	.	; APSCAN--SCAN INPUT STREAM AND BUILD NUMBERS	
9304	AF6E	.	.	; FOR AUTO PLOT	
9305	AF6E	.	.	; WHEN PROPER COLUMNS ARE FOUND, PROCESS	
9306	AF6E	.	.	; X,Y PAIRS	
9307	AF6E	.	.	; AUTO PLOT MUST BE ON WHEN CALLED	
9308	AF6E	.	.	; ENTRY--DONT CARE	
9309	AF6E	.	.	; EXIT---ALL REGISTERS DESTROYED	
9310	AF6E	.	.	*****	
9311	AF6E	.	.	APSCAN EQU \$	
9312	AF6E	21	96 FB	LXI H,APFLGS ;LOAD AUTO PLOT STATUS	
9313	AF71	46	.	MOV B,M	
9314	AF72	.	.	; CHECK SKIP COUNT TO SEE IF CHARACTER	
9315	AF72	.	.	; SHOULD BE PROCESSED OR IGNORED	
9316	AF72	21	89 FF	LXI H,ZDCHAR ;FETCH THE CHAR	
9317	AF75	4E	.	MOV C,M ;AND LEAVE IN C	
9318	AF76	2A	0A FB	LHLD SKPCNT ;FETCH LINE SKIP COUNT	
9319	AF79	7C	.	MOV A,H ;STILL SKIPPING LINES?	
9320	AF7A	B7	.	ORA A	
9321	AF7B	FA	87 AF	JM APSCN1 ;NO, DO REGULAR SCAN	
9322	AF7E	.	.	; DECREMENT SKIP COUNT EACH TIME A LF IS RECEIVED	
9323	AF7E	3E	0A .	MVI A,120 ;IS CHAR A LF?	
9324	AF80	B9	.	CMP C	
9325	AF81	C0	.	RNZ ;NO, IGNORE THE CHAR	
9326	AF82	2B	.	DCX H ;YES, UPDATE THE SKIP COUNT	
9327	AF83	22	0A FB	SHLD SKPCNT	
9328	AF86	C9	.	RET	
9329	AF87	.	.	APSCN1 EQU \$	
9330	AF87	.	.	; CHECK TO SEE IF THIS CHARACTER TERMINATES	
9331	AF87	.	.	; THE NUMBER BEING BUILT (IF THE PRECEDING	
9332	AF87	.	.	; CHARACTER WAS A TRAILING SIGN)	
9333	AF87	3E	80 .	MVI A,NEXTRM ;PRECEDING CHAR SIGN??	
9334	AF89	A0	.	ANA B	
9335	AF8A	C2	E0 B0	JNZ STOP2 ;YES,STOP,THEN RESTART SCAN	
9336	AF8D	.	.	APS005 EQU \$	
9337	AF8D	79	.	MOV A,C ;RECALL CHAR	
9338	AF8E	.	.	;LOOK FOR CHARACTERS THAT COULD START NUMBER	
9339	AF8E	.	.	; + OR - ?	
9340	AF8E	FE	2B .	CPI 530 ;+?	
9341	AF90	CA	DF AF	JZ PLUMIN	
9342	AF93	FE	2D .	CPI 550 ;-?	
9343	AF95	CA	DF AF	JZ PLUMIN	
9344	AF98	.	.	; DECIMAL POINT ?	
9345	AF98	FE	2E .	CPI 560 ;.?	
9346	AF9A	CA	11 B0	JZ PERIOD	
9347	AF9D	.	.	; DIGIT?	
9348	AF9D	FE	30 .	CPI 600 ;0?	
9349	AF9F	DA	A7 AF	JC APS010 ;NO	
9350	AFA2	FE	3A .	CPI 720 ;9?	
9351	AFA4	DA	33 B0	JC DIGIT ;YES	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 254
9352	AFA7	.	.	APS010 EQU \$	
9353	AFA7	.	.	;CANNOT START A NEW NUMBER	
9354	AFA7	.	.	; IS ONE CURRENTLY BEING BUILT??	
9355	AFA7	3E	01	MVI A,NIP ;IS A NUMBER IN PROGRESS?	
9356	AFA9	A0	.	ANA B	
9357	AFAA	C8	.	RZ ;NO,IGNORE CHARACTER	
9358	AFAB	.	.	;NUMBER IS BEING BUILT	
9359	AFAB	.	.	; ONLY LEGAL CHARACTERS NOW ARE E, \$, AND ,	
9360	AFAB	79	.	MOV A,C ;RESTORE CHAR IN A	
9361	AFAC	FE	45	CPI 105Q ;IS IT AN E?	
9362	AFAE	CA	4A	B0 JZ EXP	
9363	AFB1	FE	2C	.	
9364	AFB3	CA	6B	B0 JZ COMMA ;COMMA?	
9365	AFB6	FE	24	.	
9366	AFB8	CA	5E	B0 CPI 44Q ;\$?	
9367	AFBB	.	.	.	
9368	AFBB	.	.	.	
9369	AFBB	C3	B4	B0 JZ DOLLAR ;CHAR RECEIVED CANNOT GO INTO NUMBER	
				JMP STOP ; TERMINATE THE NUMBER BEING BUILT	

				PAGE 255	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	
9371	AFBE	.	.	.	;*****
9372	AFBE	.	.	.	; APCR--CARRIAGE RETURN RECEIVED
9373	AFBE	.	.	.	; CHECK FOR AUTO PLOT MODE, AND PROCESS IF NECESS
9374	AFBE	.	.	.	; ENTRY--DONT CARE
9375	AFBE	.	.	.	; EXIT---ALL REGISTERS DESTROYED
9376	AFBE	.	.	.	;*****
9377	AFBE	.	.	.	APCR EQU \$
9378	AFBE	3A	96	FB	LDA APFLGS ;FETCH AUTO PLOT FLAGS
9379	AFC1	47	.	.	MOV B,A ;SAVE FLAGS
9380	AFC2	.	.	.	; IF NOT IN AUTO PLOT MODE, OR NUMBER NOT BEING
9381	AFC2	.	.	.	; BUILT, IGNORE THE CR
9382	AFC2	E6	02	.	ANI APIP ;AUTO PLOT GOING?
9383	AFC4	CA	C0	00	JZ ZCRRET ;NO, PROCESS THE CR
9384	AFC7	3E	01	.	MVI A,NIP ;NUMBER BEING BUILT?
9385	AFC9	A0	.	.	ANA B
9386	AFCA	C4	B4	B0	CNZ STOP ;YES, TERMINATE IT
9387	AFCD	C3	C0	00	JMP ZCRRET ;PROCESS THE RETURN
9388	AFD0	.	.	.	;*****
9389	AFD0	.	.	.	; APLF--UPDATE 'SKIP LINES' COUNT FOR AUTO PLOT
9390	AFD0	.	.	.	; ENTRY--DONT CARE
9391	AFD0	.	.	.	; EXIT HL, A DESTROYED
9392	AFD0	.	.	.	;*****
9393	AFD0	.	.	.	APLF EQU \$
9394	AFD0	2A	0A	FB	LHLD SKPCNT ;FETCH COUNT
9395	AFD3	7C	.	.	MOV A,H ;STILL COUNTING
9396	AFD4	B7	.	.	ORA A
9397	AFD5	FA	8B	00	JM ZLNFD ;NO, DO THE LINE FEED
9398	AFD8	2B	.	.	DCX H ;YES, UPDATE COUNT
9399	AFD9	22	0A	FB	SHLD SKPCNT
9400	AFDC	C3	8B	00	JMP ZLNFD ;DO THE LINE FEED

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 256
9402	AFDF	.	.	*****	
9403	AFDF	.	.	; PLUMIN PROCESS + OR - SIGN	
9404	AFDF	.	.	; ENTRY C = CHAR	
9405	AFDF	.	.	; B = APFLGS	
9406	AFDF	.	.	*****	
9407	AFDF	.	.	PLUMIN EQU \$	
9408	AFDF	3E	20	MVI A,NMBD ;SHOULD THIS BE DIGIT?	
9409	AFE1	A0	.	ANA B	
9410	AFE2	C2	E0 B0	JNZ STOP2 ;YES, TERMINATE NUMBER	
9411	AFES	3E	40	MVI A,HAVES ;SET HAVE SIGN BIT	
9412	AFE7	CD	AF B8	CALL STAPFL	
9413	AFEA	3E	01	MVI A,NIP ;NUMBER BEING BUILT?	
9414	AFEC	A0	.	ANA B	
9415	AFED	CA	82 B0	JZ START ;NO, START ONE GOING	
9416	AFF0	.	.	; HAVE + OR - IN MIDDLE NUMBER	
9417	AFF0	.	.	; PREVIOUS CHAR MUST BE E	
9418	AFF0	2A	91 FB	LHLD NUMPTR ;POINTER TO BUFFER	
9419	AFF3	2B	.	DCX H ;POINTER TO PREVIOUS CHAR	
9420	AFF4	7E	.	MOV A,M ;WAS IT AN E	
9421	AFF5	FE	45	CPI 1050	
9422	AFF7	CA	A3 B0	JZ STORE ;YES, SIGN IS OK	
9423	AFFA	.	.	*****	
9424	AFFA	.	.	; ROM BREAK 7	
9425	AFFA	C3	02 B0	JMP ZBRK7C	
9426	AFFD	.	.	ORG ZBRK6+40000	
9427	B000	.	.	ZBRK7 EQU \$	
9428	B000	54	.	DB VERSN	
9429	B001	B0	.	DB ZBRK7/256	
9430	B002	.	.	ZBRK7C EQU \$	
9431	B002	.	.	*****	
9432	B002	.	.	; TEST FOR TRAILING SIGN (COBOL PROTOCOL)	
9433	B002	.	.	; NOT OK IF HAVE SIGN ALREADY, OR IN EXPONENTIAL	
9434	B002	.	.	; FIELD	
9435	B002	3E	44	MVI A,HAVES+HAVEE ;SIGN OR E FIELD ALREAD	
9436	B004	A0	.	ANA B	
9437	B005	C2	E0 B0	JNZ STOP2 ;YES, NOT TRAILING SIGN	
9438	B008	.	.	; INSERT TRAILING SIGN AT START OF STRING	
9439	B008	21	0D FB	LXI H,NUMBUF ;START OF STRING	
9440	B00B	71	.	MOV M,C ;STORE THE SIGN	
9441	B00C	.	.	; NEXT CHARACTER RECEIVED TERMINATES THE NUMBER	
9442	B00C	.	.	; BEING BUILT, NO MATTER WHAT IT IS	
9443	B00C	3E	80	MVI A,NEXTRM ;SET TERMINATE FLAG	
9444	B00E	C3	AF B8	JMP STAPFL	

				PAGE 257	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	
9446	B011	.	.	.	;*****
9447	B011	.	.	.	; PERIOD PROCESS DECIMAL POINT
9448	B011	.	.	.	; ENTRY C = CHAR
9449	B011	.	.	.	; B = APFLGS
9450	B011	.	.	.	;*****
9451	B011	.	.	.	PERIOD EQU \$
9452	B011	3E	20	.	MVI A,NMBD ;SHOULD THIS BE DIGIT?
9453	B013	A0	.	.	ANA B
9454	B014	C2	E0	B0	JNZ STOP2 ;YES, TERMINATE NUMBER
9455	B017	3E	01	.	MVI A,NIP ;IN MIDDLE OF NUMBER?
9456	B019	A0	.	.	ANA B
9457	B01A	C2	25	B0	JNZ PRD010 ;YES
9458	B01D	.	.	.	; START A NEW NUMBER WITH A DECIMAL POINT
9459	B01D	3E	08	.	MVI A,HAVEP ;HAVE FOUND A .
9460	B01F	CD	AF	B8	CALL STAPFL
9461	B022	C3	82	B0	JMP START ;START BUILDING NUMBER
9462	B025	.	.	.	PRD010 EQU \$
9463	B025	.	.	.	;HAVE FOUND DECIMAL POINT IN MIDDLE OF NUMBER
9464	B025	3E	0C	.	MVI A,HAVEP+HAVEE ;ALREADY HAVE, OR IN
9465	B027	A0	.	.	ANA B ;EXPONENT FIELD??
9466	B028	C2	E0	B0	JNZ STOP2 ;YES, TERMINATE
9467	B028	3E	08	.	MVI A,HAVEP ;NO, SET FLAG
9468	B02D	CD	AF	B8	CALL STAPFL
9469	B030	C3	A3	B0	JMP STORE ;STORE IT

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 258
=====
9471      B033      . . .      ;*****
9472      B033      . . .      ; DIGIT  PROCESS 0-9
9473      B033      . . .      ; ENTRY  C = CHAR
9474      B033      . . .      ;          B = APFLGS
9475      B033      . . .      ;*****
9476      B033      . . .      DIGIT EQU $
9477      B033      . . .      ; CHECK IGNORE COUNT FOR DIGIT ENTERED LOCALLY
9478      B033      . . .      ; FROM KEYBOARD BEFORE AUTOPLLOT STARTED
9479      B033      21 05 FB    LXI  H,IGNCNT ;FETCH THE COUNT
9480      B036      35 . .     DCR  M        ;STILL IGNORING DIGITS?
9481      B037      F0 . .     RP          ;YES, IGNORE THIS ONE
9482      B038      34 . .     INR  M        ;NO, RESTORE COUNT
9483      B039      3E 20 .     MVI  A,NMBD  ;CLEAR NEED DIGIT FLAG
9484      B03B      CD B5 B8    CALL CLAPFL
9485      B03E      F6 10 .     ORI  HAVED   ;SET FOUND A DIGIT FLAG
9486      B040      77 . .     MOV  M,A
9487      B041      3E 01 .     MVI  A,NIP   ;NUMBER BEING BUILT?
9488      B043      A0 . .     ANA  B
9489      B044      CA 82 B0    JZ   START   ;NO, START ONE GOING
9490      B047      C3 A3 B0    JMP  STORE   ;YES, STORE IT
9491      B04A      . . .      ;*****
9492      B04A      . . .      ; EXP--PROCESS CAP E
9493      B04A      . . .      ; ENTRY  C = CHAR
9494      B04A      . . .      ;          B = APFLGS
9495      B04A      . . .      ;*****
9496      B04A      . . .      EXP EQU $
9497      B04A      3E 20 .     MVI  A,NMBD  ;SHOULD THIS BE DIGIT?
9498      B04C      A0 . .     ANA  B
9499      B04D      C2 B4 B0    JNZ  STOP    ;YES, TERMINATE
9500      B050      3E 04 .     MVI  A,HAVEE ;ALREADY GOT ONE?
9501      B052      A0 . .     ANA  B
9502      B053      C2 B4 B0    JNZ  STOP    ;YES, TERMINATE
9503      B056      3E 04 .     MVI  A,HAVEE ;E IS VALID
9504      B058      CD AF B8    CALL STAPFL  ;SET GOT ONE FLAG
9505      B05B      C3 A3 B0    JMP  STORE   ;SAVE IT
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 259
=====
9507      B05E      . . .      ;*****
9508      B05E      . . .      ; DOLLAR PROCESS $
9509      B05E      . . .      ; ENTRY C = CHAR
9510      B05E      . . .      ;          B = APFLGS
9511      B05E      . . .      ;*****
9512      B05E      . . .      DOLLAR EQU $
9513      B05E      3E 20 .      MVI A,NMBD      ;SHOULD THIS BE A DIGIT?
9514      B060      A0 . . .      ANA B
9515      B061      C2 B4 B0      JNZ STOP        ;YES, TERMINATE
9516      B064      3E 1C .      MVI A,HAVED+HAVEP+HAVEE ;IN WRONG PLACE?
9517      B066      A0 . . .      ANA B
9518      B067      C2 B4 B0      JNZ STOP        ;YES, TERMINATE
9519      B06A      C9 . . .      RET            ;IGNORE IT
9520      B06B      . . .      ;*****
9521      B06B      . . .      ; COMMA PROCESS COMMA
9522      B06B      . . .      ; IMBEDDED COMMA IS OK IF NEXT VALUE IS DIGIT
9523      B06B      . . .      ; CANNOT BE IN E FIELD
9524      B06B      . . .      ; ENTRY C = CHAR
9525      B06B      . . .      ;          B = APFLGS
9526      B06B      . . .      ;*****
9527      B06B      . . .      COMMA EQU $
9528      B06B      3E 20 .      MVI A,NMBD      ;SHOULD THIS BE DIGIT?
9529      B06D      A0 . . .      ANA B
9530      B06E      C2 B4 B0      JNZ STOP        ;YES, TERMINATE
9531      B071      . . .      ;HAS A DIGIT BEEN FOUND YET?
9532      B071      3E 10 .      MVI A,HAVED
9533      B073      A0 . . .      ANA B
9534      B074      CA B4 B0      JZ STOP         ;NO, IN WRONG PLACE
9535      B077      . . .      ; AFTER . OR IN E FIELD?
9536      B077      3E 0C .      MVI A,HAVEE+HAVEP
9537      B079      A0 . . .      ANA B
9538      B07A      C2 B4 B0      JNZ STOP        ;YES TERMINATE
9539      B07D      . . .      ; COMMA CAN BE IGNORED IF NEXT CHAR IS DIGIT
9540      B07D      3E 20 .      MVI A,NMBD
9541      B07F      C3 AF B8      JMP STAPFL
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 260
9543	B082	. . .	;*****	
9544	B082	. . .	; START--START BUILDING A NUMERICAL STRING FROM	
9545	B082	. . .	; INPUT CHARACTERS	
9546	B082	. . .	;*****	
9547	B082	. . .	START EQU \$	
9548	B082	3E 2C .	MVI A,HAVEE+HAVEP+NMBD ;CLEAR FLAGS	
9549	B084	CD B5 B8	CALL CLAPFL	
9550	B087	F6 01 .	ORI NIP ;SET STARTING NEW NUMBER FLA	
9551	B089	77 . .	MOV M,A	
9552	B08A	21 0D FB	LXI H,NUMBUF ;STORE SPACE FOR	
9553	B08D	36 20 .	MVI M,40Q ;POSSIBLE INSERTION OF	
9554	B08F	23 . .	INX H ;TRAILING SIGN	
9555	B090	22 91 FB	SHLD NUMPTR ;UPDATE BUFFER POINTER	
9556	B093	AF . .	XRA A	
9557	B094	32 93 FB	STA NUMLEN ;RESET NUMBER LENGTH	
9558	B097	. . .	;STORE CURSOR LOCATION TO ALLOW ENHANCEMENT OF	
9559	B097	. . .	;FIRST DIGIT IN NUMBER	
9560	B097	2A C0 FF	LHLD ZCUROW ;FETCH CURSOR ROW & COLUMN	
9561	B09A	CD 23 B9	CALL ADJCOL ;FIX COL IF PLOT FROM DISPLA	
9562	B09D	. . .	; CONVERT TO ABSOLUTE ROW POSITION	
9563	B09D	CD DF B1	CALL ABSROW	
9564	B0A0	22 94 FB	SHLD BGNCUR ;SAVE THEM	
9565	B0A3	. . .	; FALL INTO STORE ROUTINE	
9566	B0A3	. . .	;*****	
9567	B0A3	. . .	; STORE--STORE ASCII CHAR INTO NUMERICAL STRING	
9568	B0A3	. . .	; BUFFER. DONT GO BEYOND END	
9569	B0A3	. . .	; ENTRY C = CHAR	
9570	B0A3	. . .	;*****	
9571	B0A3	. . .	STORE EQU \$	
9572	B0A3	. . .	; SEE IF ANY ROOM LEFT	
9573	B0A3	21 93 FB	LXI H,NUMLEN ;CURRENT STRING LENGTH	
9574	B0A6	7E . .	MOV A,M	
9575	B0A7	FE 14 .	CPI MAXLEN ;TOO BIG?	
9576	B0A9	D0 . .	RNC ;YES, DONT STORE	
9577	B0AA	34 . .	INR M ; NO, INCRMENT LENGTH	
9578	B0AB	2A 91 FB	LHLD NUMPTR ;POINTER TO EMPTY SLOT	
9579	B0AE	71 . .	MOV M,C ;STORE THE CHAR	
9580	B0AF	23 . .	INX H ;BUMP POINTER	
9581	B0B0	22 91 FB	SHLD NUMPTR	
9582	B0B3	C9 . .	RET	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 261
9584	B0B4	. . .	;*****	
9585	B0B4	. . .	; STOP--STOP BUILDING NUMBER	
9586	B0B4	. . .	; PROCESS DATA COLUMN IF NECESSARY	
9587	B0B4	. . .	; ENTRY B = APFLGS	
9588	B0B4	. . .	;*****	
9589	B0B4	. . .	STOP EQU \$	
9590	B0B4	. . .	; CLEAR PERTINENT AUTOPLLOT FLAGS	
9591	B0B4	3E FD .	MVI A,-1-APIP	
9592	B0B6	CD B5 B8	CALL CLAPFL	
9593	B0B9	. . .	; SEE IF ANY DIGITS WERE FOUND	
9594	B0B9	3E 10 .	MVI A,HAVED ;ANY DIGITS?	
9595	B0B8	A0 . .	ANA B	
9596	B0BC	C8 . .	RZ ;NO, DISREGARD NUMBER BUILT	
9597	B0BD	. . .	;STORE TERMINATOR	
9598	B0BD	2A 91 FB	LHLD NUMPTR	
9599	B0C0	36 20 .	MVI M,400 ;STORE A SPACE	
9600	B0C2	. . .	; FETCH COLUMN COUNT, SEE IF THIS IS THE X OR	
9601	B0C2	. . .	; Y DATA COLUMN	
9602	B0C2	3A 0C FB	LDA COLCNT	
9603	B0C5	3C . .	INR A ;CURRENT COLUMN	
9604	B0C6	. . .	; SEE IF THIS IS THE X COL	
9605	B0C6	21 C2 FB	LXI H,XCOLBF ;POINTER TO X COLUMN VALUE	
9606	B0C9	BE . .	CMP M ;SAME?	
9607	B0CA	CC EC B0	CZ COLX ;YES, PROCESS X COLUMN	
9608	B0CD	. . .	; SEE IF THIS IS THE Y COLUMN	
9609	B0CD	21 C1 FB	LXI H,YCOLBF ;POINTER TO Y COLUMN VALUE	
9610	B0D0	BE . .	CMP M ;SAME?	
9611	B0D1	CC 57 B1	CZ COLY ;YES, PROCESS Y COLUMN	
9612	B0D4	. . .	; SEE IF THIS IS THE LAST COLUMN	
9613	B0D4	21 C3 FB	LXI H,NMCLBF ;POINTER TO NUMBER OF COLUMN	
9614	B0D7	BE . .	CMP M	
9615	B0D8	DA DC B0	JC STP010 ;NO, STILL TOO SMALL	
9616	B0DB	AF . .	XRA A ;YES, RESET COLUMN COUNTER	
9617	B0DC	. . .	STP010 EQU \$	
9618	B0DC	32 0C FB	STA COLCNT ;STORE NEW COLUMN COUNT	
9619	B0DF	C9 . .	RET	
9620	B0E0	. . .	;*****	
9621	B0E0	. . .	; STOP2--TERMINATE NUMBER BEING BUILT, BUT USE	
9622	B0E0	. . .	; TERMINATOR AS POSSIBLE FIRST CHAR OF NEW	
9623	B0E0	. . .	; NUMBER	
9624	B0E0	. . .	;*****	
9625	B0E0	. . .	STOP2 EQU \$	
9626	B0E0	C5 . .	PUSH B ;SAVE CHAR IN C	
9627	B0E1	CD B4 B0	CALL STOP ;PROCESS THE CHARACTER	
9628	B0E4	C1 . .	POP B ;RECALL CHAR	
9629	B0E5	21 89 FF	LXI H,ZDCHAR	
9630	B0E8	71 . .	MOV M,C ;USE THE SAME CHAR AGAIN	
9631	B0E9	C3 6E AF	JMP APSCAN ;REDO SCAN WITH SAME CHAR	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 262
=====
9633  B0EC      .      .      .      ;*****
9634  B0EC      .      .      .      ; COLX--HAVE DATA FROM X COLUMN IN NUMBUF
9635  B0EC      .      .      .      ; 0. HIGHLIGHT NUMBER ON DISPLAY IN INVERSE VIDEO
9636  B0EC      .      .      .      ; 1. CONVERT TO FLOATING POINT
9637  B0EC      .      .      .      ; 2. SUBTRACT XMIN
9638  B0EC      .      .      .      ; 3. MULTIPLY BY SCALE FACTOR
9639  B0EC      .      .      .      ; 4. CONVERT TO INTEGER
9640  B0EC      .      .      .      ; 5. PLOT IF Y IS ALSO READY
9641  B0EC      .      .      .      ;
9642  B0EC      .      .      .      ; X COORD = (X-XMIN) * XSCALE
9643  B0EC      .      .      .      ;*****
9644  B0EC      .      .      .      COLX EQU $
9645  B0EC      F5      .      .      PUSH PSW          ;SAVE A
9646  B0ED      CD      86  B1      CALL HILITE       ;HILITE NUMBER IN I.V.
9647  B0F0      .      .      .      ;CONVERT TO FLOATING POINT COORDINATE
9648  B0F0      CD      35  B1      CALL GETFP        ;CONVERT STRING TO FLOAT PNT
9649  B0F3      21      BC  FB      LXI H,XMINBF     ;POINTER TO XMIN
9650  B0F6      CD      D3  BD      CALL SB          ;HAVE X-XMIN NOW
9651  B0F9      21      A2  90      LXI H,XSCALE     ;MULTIPLY BY SCALE FACTOR
9652  B0FC      CD      8C  BD      CALL MUL
9653  B0FF      DA      19  B1      JC CLX005        ;EXIT IF OVERFLOW
9654  B102      CD      C9  B8      CALL INT         ;CONVERT TO INTEGER
9655  B105      11      46  00      LXI D,XOFSET     ;ADD OFFSET FOR FRAME
9656  B108      19      .      .      DAD D
9657  B109      .      .      .      ; NOW HAVE X COORD IN HL
9658  B109      22      DA  90      SHLD XNEW        ;SAVE IT
9659  B10C      .      .      .      ; NOW SEE IF A VECTOR CAN BE DRAWN
9660  B10C      3A      97  FB      LDA APFLG2       ;HAVE Y COORD YET?
9661  B10F      E6      08  .      ANI HAVEY
9662  B111      C2      1B  B1      JNZ CLX010       ;YES, DRAW VECTOR
9663  B114      .      .      .      ; CANT PLOT YET, DONT HAVE Y
9664  B114      3E      04  .      MVI A,HAVEX     ;SET FLAG FOR X
9665  B116      CD      BC  B8      CALL STAPF2
9666  B119      .      .      .      CLX005 EQU $
9667  B119      F1      .      .      POP PSW
9668  B11A      C9      .      .      RET              ;DONE
9669  B11B      .      .      .      ;HAVE BOTH X AND Y, CAN DRAW A VECTOR
9670  B11B      .      .      .      CLX010 EQU $
9671  B11B      3E      0C  .      MVI A,HAVEX+HAVEY ;CLEAR FLAGS
9672  B11D      CD      C2  B8      CALL CLAPF2
9673  B120      CD      DB  65      CALL VECTOR      ;DRAW THE VECTOR
9674  B123      .      .      .      ; CHECK POINT COUNT SPECIFIED. STOP AUTO PLOT
9675  B123      .      .      .      ; WHEN NUMBER SPECIFIED HAS BEEN REACHED
9676  B123      2A      08  FB      LHL D,PNTCNT    ;FETCH POINT COUNT
9677  B126      7C      .      .      MOV A,H         ;WAS A COUNT SPECIFIED?
9678  B127      B7      .      .      ORA A           ;IF -, NO COUNT WAS SPECIFIE
9679  B128      FA      33  B1      JM CLX030       ;IGNORE
9680  B12B      B5      .      .      ORA L           ;HAS COUNT REACHED 0 YET?
9681  B12C      CC      82  B8      CZ APLTOF      ;YES, STOP AUTO PLOT
9682  B12F      2B      .      .      DCX H          ;UPDATE POINT COUNT
=====

```

=====					PAGE 263	
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	
=====						
9683	B130	22	08	FB	SPLD PNTCNT	
9684	B133	.	.	.	CLX030 EQU \$	
9685	B133	F1	.	.	POP PSW	
9686	B134	C9	.	.	RET	
9687	B135	.	.	.	*****	
9688	B135	.	.	.	; GETFP--CONVERT ASCII STRING TO FLOATING POINT.	
9689	B135	.	.	.	; EXIT F.P. ACCUMULATOR CONTAINS VALUE	
9690	B135	.	.	.	*****	
9691	B135	.	.	.	GETFP EQU \$	
9692	B135	.	.	.	; MUST DELETE POSSIBLE LEADING SPACE (LEFT	
9693	B135	.	.	.	; OPEN FOR INSERTION OF TRAILING SIGN)	
9694	B135	21	0D	FB	LXI H,NUMBUF ;START OF STRING	
9695	B138	7E	.	.	MOV A,M ;FETCH FIRST CHAR	
9696	B139	FE	20	.	CPI 40Q ;IS IT A SPACE?	
9697	B138	C2	3F	B1	JNZ FPINP ;NO, DO CONVERSION	
9698	B13E	23	.	.	INX H ;YES, POINT TO NEXT CHAR	
9699	B13F	.	.	.	*****	
9700	B13F	.	.	.	; FPINP--CONVERT ASCII STRING TO FLOATING	
9701	B13F	.	.	.	; POINT, TEST FOR OVER/UNDER FLOW	
9702	B13F	.	.	.	; ENTRY HL = POINTER TO STRING	
9703	B13F	.	.	.	*****	
9704	B13F	.	.	.	FPINP EQU \$	
9705	B13F	CD	4A	BB	CALL INP ;DO THE CONVERSION	
9706	B142	21	4F	B1	LXI H,FPONE ;MULTIPLY BY 1 TO TEST FOR	
9707	B145	CD	8C	BD	CALL MUL ;OVER/UNDER FLOW	
9708	B148	D0	.	.	RNC ;EXIT IF NO OVERFLOW	
9709	B149	21	53	B1	LXI H,MAXFP ;OVERFLOW, SUBTITUTE MAX VAL	
9710	B14C	C3	6E	BD	JMP LOD ;= 1E30	
9711	B14F	.	.	.	;	
9712	B14F	81	00	00	FPONE DB 201Q,0Q,0Q,0Q ;1 IN FLOATING POINT	
9713	B153	E4	49	F2	MAXFP DB 344Q,111Q,362Q,315Q ;1E30	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 264
9715	B157	.	. ;*****	
9716	B157	.	. ; COLY--HAVE DATA FROM Y COLUMN IN NUMBUF	
9717	B157	.	. ; 0. HIGHLIGHT NUMBER ON DISPLAY IN INVERSE VIDEO	
9718	B157	.	. ; 1. CONVERT TO FLOATING POINT	
9719	B157	.	. ; 2. SUBTRACT YMIN	
9720	B157	.	. ; 3. MULTIPLY BY SCALE FACTOR	
9721	B157	.	. ; 4. CONVERT TO INTEGER	
9722	B157	.	. ; 5. PLOT IF X IS ALSO READY	
9723	B157	.	. ;	
9724	B157	.	. ; Y COORD = (Y-YMIN) * YSCALE	
9725	B157	.	. ;*****	
9726	B157	.	. COLY EQU \$	
9727	B157	F5	. PUSH PSW ;SAVE A	
9728	B158	CD 86	B1 CALL HILITE ;HILITE NUMBER IN I.V.	
9729	B158	.	. ;CONVERT TO FLOATING POINT COORDINATE	
9730	B158	CD 35	B1 CALL GETFP ;CONVERT STRING TO FLOAT PNT	
9731	B15E	21 84	FB LXI H,YMINBF ;POINTER TO YMIN	
9732	B161	CD D3	BD CALL SB ;HAVE Y-YMIN NOW	
9733	B164	21 9E	90 LXI H,YSCALE ;MULTIPLY BY SCALE FACTOR	
9734	B167	CD 8C	BD CALL MUL	
9735	B16A	DA 84	B1 JC CLY010 ;EXIT IF OVERFLOW	
9736	B16D	CD C9	B8 CALL INT ;CONVERT TO INTEGER	
9737	B170	11 2D	00 LXI D,YOFSET ;ADD OFFSET FOR FRAME	
9738	B173	19	. DAD D	
9739	B174	.	. ; NOW HAVE Y COORDINATE IN HL	
9740	B174	22 D8	90 SHLD YNEW ;SAVE IT	
9741	B177	.	. ; NOW SEE IF A VECTOR CAN BE DRAWN	
9742	B177	3A 97	FB LDA APFLG2 ;HAVE X COORD YET?	
9743	B17A	E6 04	. ANI HAVEX	
9744	B17C	C2 1B	B1 JNZ CLX010 ;YES, DRAW VECTOR	
9745	B17F	.	. ; CANT PLOT YET, DONT HAVE X	
9746	B17F	3E 08	. MVI A,HAVEY ;SET FLAG FOR Y	
9747	B181	CD BC	B8 CALL STAPF2	
9748	B184	.	. CLY010 EQU \$	
9749	B184	F1	. POP PSW	
9750	B185	C9	. RET	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 265
9752	B186	. . .	*****	
9753	B186	. . .	; HILITE-- TURN INVERSE VIDEO ON AROUND NUMBER	
9754	B186	. . .	; ON DISPLAY	
9755	B186	. . .	; ENTRY BGNCUR = CURSOR POSITION OF FIRST CHAR	
9756	B186	. . .	; ZCUROW = CURSOR POS. OF TERMINATING CHAR	
9757	B186	. . .	*****	
9758	B186	. . .	HILITE EQU \$	
9759	B186	2A C0 FF	LHLD ZCUROW ;SAVE CURSOR ROW, COL	
9760	B189	E5 . .	PUSH H	
9761	B18A	CD 23 B9	CALL ADJCOL ;ADJUST CURSOR COLUMN	
9762	B18D	22 C0 FF	SHLD ZCUROW ;PUT CURSOR THERE	
9763	B190	E5 . .	PUSH H ;SAVE ENDING POSITION	
9764	B191	. . .	; SEE IF IVOFF IS AT END OF SCREEN LINE	
9765	B191	. . .	; IF SO, IT ISNT NEEDED, DONT TRY TO PUT IT IN	
9766	B191	7C . .	MOV A,H ;CHECK COLUMN	
9767	B192	FE 50 .	CPI MAXCOL+1 ;AT END OF LINE?	
9768	B194	D2 9B B1	JNC HIL005 ;YES, DONT ADD IVOFF CODE	
9769	B197	AF . .	XRA A ;CODE TO TURN I.V. OFF	
9770	B198	CD BA 00	CALL ZDSPC0 ;TURN INVERSE VIDEO OFF	
9771	B19B	. . .	HIL005 EQU \$	
9772	B19B	2A 94 FB	LHLD BGNCUR ;FETCH STARTING CURSOR POS	
9773	B19E	CD E6 B1	CALL RELROW ;CONVERT TO RELATIVE POS	
9774	B1A1	7D . .	MOV A,L ;SEE IF ROW ON SCREEN	
9775	B1A2	B7 . .	ORA A ;.LT. 0 ?	
9776	B1A3	FA DB B1	JM HIL030 ;YES, DONE	
9777	B1A6	FE 18 .	CPI 24 ;.GT. 23 ?	
9778	B1A8	D2 DB B1	JNC HIL030 ;YES, DONE	
9779	B1AB	22 C0 FF	SHLD ZCUROW ;PUT CURSOR THERE	
9780	B1AE	E5 . .	PUSH H ;SAVE STARTING POSITION	
9781	B1AF	3E 02 .	MVI A,2Q ;DISPLAY CODE FOR IV ON	
9782	B1B1	CD BA 00	CALL ZDSPC0 ;ADD INVERSE VIDEO CODE	
9783	B1B4	. . .	; IF BEGINNING AND ENDING ROWS OF NUMBER AREN'T	
9784	B1B4	. . .	; THE SAME, GO BACK AND PUT AN INVERSE VIDEO	
9785	B1B4	. . .	; CODE AT THE START OF EACH ROW BETWEEN THEM.	
9786	B1B4	. . .	; LEFT MARGIN IS THE STARING COLUMN OF THE ROW	
9787	B1B4	. . .	HIL010 EQU \$	
9788	B1B4	E1 . .	POP H ;HL=STARTING ROW, COL	
9789	B1B5	D1 . .	POP D ;DE=ENDING ROW,COL	
9790	B1B6	7D . .	MOV A,L ;SEE IF START ROW=END ROW	
9791	B1B7	BB . .	CMP E ;ARE THEY EQUAL?	
9792	B1B8	D2 D6 B1	JNC HIL020 ;YES, DONE	
9793	B1B8	. . .	;MOVE DOWN ONE ROW	
9794	B1B8	3C . .	INR A	
9795	B1BC	6F . .	MOV L,A ;STORE NEW STARTING ROW	
9796	B1BD	. . .	; ARE THE STARTING AND ENDING ROWS NOW THE SAME??	
9797	B1BD	BB . .	CMP E	
9798	B1BE	3A BF FF	LDA ZLFTMG ;(LOAD LOC OF NEW IV CODE)	
9799	B1C1	C2 C8 B1	JNZ HIL015 ;NO,ADD IV CODE TO ST OF ROW	
9800	B1C4	. . .	; ABOUT TO ADD IV CODE TO SAME LINE THAT CONTAINS	
9801	B1C4	. . .	; THE IVOFF CODE PREVIOUSLY STORED	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 266
9802	B1C4	. . .	; DO NOT ADD AN IV CODE IF THE ENDING COL IS THE	
9803	B1C4	. . .	; SAME AS THE LEFT MARGIN. AN IV CODE PUT	
9804	B1C4	. . .	; THERE WILL WIPE OUT THE IV OFF CODE	
9805	B1C4	8A . .	CMP D ;SAME COL AS IVOFF CODE?	
9806	B1C5	CA D6 B1	JZ HIL020 ;YES, DONE	
9807	B1C8	. . .	HIL015 EQU \$	
9808	B1C8	. . .	; ADD AN IV ON CODE TO START OF THIS LINE	
9809	B1C8	67 . .	MOV H,A ;STORE STARTING COLUMN	
9810	B1C9	D5 . .	PUSH D ;STORE STARTING ROW,COL	
9811	B1CA	E5 . .	PIJSH H ;SAVE ENDING ROW,COL	
9812	B1CB	22 C0 FF	SHLD ZCURROW ;SET ROW AND COL	
9813	B1CE	3E 02 .	MVI A,2 ;CODE FOR INVERSE VIDEO	
9814	B1D0	CD BA 00	CALL ZDSPCO ;ADD INVERSE VIDEO	
9815	B1D3	C3 B4 B1	JMP HIL010 ;GO THRU AGAIN	
9816	B1D6	. . .	HIL020 EQU \$	
9817	B1D6	E1 . .	POP H ;PUT CURSOR BACK WHERE IT WA	
9818	B1D7	22 C0 FF	SHLD ZCURROW	
9819	B1DA	C9 . .	RET	
9820	B1DB	. . .	HIL030 EQU \$	
9821	B1DB	E1 . .	POP H ;RESTORE STACK	
9822	B1DC	C3 D6 B1	JMP HIL020 ;TERMINATE	
9823	B1DF	. . .	;*****	
9824	B1DF	. . .	; ABSROW--CONVERT RELATIVE SCREEN ROW TO ABSOLUTE	
9825	B1DF	. . .	; ENTRY L = REL ROW	
9826	B1DF	. . .	; EXIT L = ABS ROW	
9827	B1DF	. . .	; DE, A DESTROYED	
9828	B1DF	. . .	;*****	
9829	B1DF	. . .	ABSROW EQU \$	
9830	B1DF	11 A3 FF	LXI D,ZTLIN0 ;ABSOLUTE ROW FOR TOP LINE	
9831	B1E2	1A . .	LDAX D	
9832	B1E3	85 . .	ADD L ;ADD RELATIVE POS ON SCREEN	
9833	B1E4	6F . .	MOV L,A ;RESTORE L	
9834	B1E5	C9 . .	RET	
9835	B1E6	. . .	;*****	
9836	B1E6	. . .	; RELROW CONVERT ABSOLUTE ROW TO RELATIVE ROW	
9837	B1E6	. . .	; ON SCREEN	
9838	B1E6	. . .	; ENTRY L = ABSOLUTE ROW	
9839	B1E6	. . .	; EXIT L = REL. SCREEN ROW, C,DE,A DESTROYED	
9840	B1E6	. . .	;*****	
9841	B1E6	. . .	RELROW EQU \$	
9842	B1E6	11 A3 FF	LXI D,ZTLIN0 ;ABSOLUTE ROW FOR TOP LINE	
9843	B1E9	1A . .	LDAX D	
9844	B1EA	4F . .	MOV C,A ;SAVE IN C	
9845	B1EB	7D . .	MOV A,L ;A = CURRENT ABSOLUTE ROW	
9846	B1EC	91 . .	SUB C ;DIFF BETWEEN TOP AND CURREN	
9847	B1ED	6F . .	MOV L,A ;RESTORE L	
9848	B1EE	C9 . .	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 267
9850	B1EF	.	.	*****	
9851	B1EF	.	.	; APAXES--DRAW AUTO PLOT FRAME AND AXES, ADD TIC	
9852	B1EF	.	.	; MARKS, AND LABEL TICS	
9853	B1EF	.	.	*****	
9854	B1EF	.	.	APAXES EQU \$	
9855	B1EF	CD	69 B9	CALL RNGCHK ;MIN,MAX VALUES OK?	
9856	B1F2	F2	80 B9	JP APERR ;NO, REPORT ERROR	
9857	B1F5	CD	F2 B9	CALL TICCHK ;NO. OF TICS OK??	
9858	B1F8	F2	4E BA	JP TICERR ;NO, REPORT ERROR	
9859	B1FB	CD	76 BA	CALL LBLCHK ;LABELS OK ?	
9860	B1FE	F2	D0 BA	JP LBLERR ;NO, REPORT ERROR	
9861	B201	CD	1E 76	CALL GTXOF1 ;G TEXT OFF	
9862	B204	CD	02 70	CALL TGCOF1 ;CURSOR OFF	
9863	B207	CD	73 AA	CALL APINIT ;GET SCALE FACTORS	
9864	B20A	CD	D9 AA	CALL APMUOF ;AUTO PLOT MENU OFF	
9865	B20D	CD	AC 60	CALL GVON1 ;TURN GRAPHICS ON	
9866	B210	3E	02 .	MVI A,2 ;WANT SET MODE	
9867	B212	CD	20 72	CALL SETMD1	
9868	B215	3E	FF .	MVI A,3770 ;SET SOLID PATTERN	
9869	B217	32	C7 FB	STA PAT2 ;FOR HLINE	
9870	B21A	CD	86 B2	CALL FRAME ;DRAW BASIC FRAME	
9871	B21D	CD	80 B2	CALL XAXIS ;DRAW X AXIS	
9872	B220	CD	DC B3	CALL YAXIS ;DRAW Y AXIS	
9873	B223	3E	01 .	MVI A,SLANT ;FOR TIC LABELS:	
9874	B225	CD	60 A2	CALL CLFLG6 ;TURN SLANT OFF	
9875	B228	AF	. .	XRA A	
9876	B229	32	DA FB	STA TXMAG ;SET TO SMALLEST SIZE	
9877	B22C	CD	48 76	CALL ANGLE ;SET TO NORMAL ANGLE	
9878	B22F	.	.	; DRAW UNLABELED X TICS	
9879	B22F	21	A8 FB	LXI H,XTICBF ;POINTER TO X TIC SPACING	
9880	B232	22	EC FA	SHLD TICPTR	
9881	B235	3E	80 .	MVI A,TICLBL ;CLEAR LABEL FLAG	
9882	B237	CD	C2 B8	CALL CLAPF2	
9883	B23A	CD	F6 B2	CALL XTICS	
9884	B23D	.	.	; DRAW UNLABELED Y TICS	
9885	B23D	21	A0 FB	LXI H,YTICBF ;Y TIC SPACING	
9886	B240	22	EC FA	SHLD TICPTR	
9887	B243	CD	23 B4	CALL YTICS	
9888	B246	3E	02 .	MVI A,APLABL ;SET AUTO PLOT LABEL IN	
9889	B248	CD	67 A2	CALL STFLG7 ;PROGRESS FLAG	
9890	B248	.	.	; DRAW LABELED X TICS	
9891	B248	21	AC FB	LXI H,XLBLBF ;X LABEL SPACING	
9892	B24E	22	EC FA	SHLD TICPTR	
9893	B251	3E	80 .	MVI A,TICLBL ;SET LABEL FLAG	
9894	B253	CD	BC B8	CALL STAPF2	
9895	B256	3E	05 .	MVI A,5 ;SET LORG FOR TOP, CENTER	
9896	B258	CD	59 99	CALL LORG1	
9897	B25B	21	A7 F9	LXI H,XLBASC ;SCAN LABEL STRING FOR FORMA	
9898	B25E	CD	62 B6	CALL LBLFMT	
9899	B261	CD	F6 B2	CALL XTICS ;DRAW LABELED TICS	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 268
9900	B264	.	.	; DRAW LABELED Y TICS	
9901	B264	21	A4	FB LXI H,YLBLBF ;Y LABEL SPACING	
9902	B267	22	EC	FA SHLD TICPTR	
9903	B26A	3E	07	. MVI A,7 ;SET LORG FOR RT JUST, MID	
9904	B26C	CD	59	99 CALL LORG1	
9905	B26F	21	4F	F9 LXI H,YLBASC ;SCAN LABEL STRING FOR FORMA	
9906	B272	CD	62	B6 CALL LBLFMT	
9907	B275	CD	23	B4 CALL YTICS ;DRAW LABELLED TICS	
9908	B278	3E	02	. MVI A,APLABL ;CLEAR AUTO PLOT LABEL IN	
9909	B27A	CD	6D	A2 CALL CLFLG7 ;PROGRESS FLAG	
9910	B27D	3E	01	. MVI A,MOVE ;LIFT THE PEN IN CASE	
9911	B27F	CD	26	A2 CALL STFLG1 ;AUTO PLOT IS ON	
9912	B282	AF	.	. XRA A ;BACK TO NORMAL LORG	
9913	B283	C3	59	99 JMP LORG1	
9914	B286	.	.	. ;*****	
9915	B286	.	.	. ; FRAME--DRAW BOX AROUND PLOTTING AREA	
9916	B286	.	.	. ; XOFSET,YOFSET IS DISTANCE TO LL CORNER	
9917	B286	.	.	. ; XAXLEN,YAXLEN IS LENGTH OF SIDES	
9918	B286	.	.	. ;*****	
9919	B286	.	.	. FRAME EQU \$	
9920	B286	11	46	00 LXI D,XOFSET	
9921	B289	21	2D	00 LXI H,YOFSET	
9922	B28C	01	99	FD LXI B,-XAXLEN-1	
9923	B28F	CD	06	B5 CALL HLINE ;DRAW BOTTOM	
9924	B292	11	46	00 LXI D,XOFSET	
9925	B295	21	5D	01 LXI H,YOFSET+YAXLEN	
9926	B298	CD	06	B5 CALL HLINE ;DRAW TOP	
9927	B29B	11	46	00 LXI D,XOFSET	
9928	B29E	21	2D	00 LXI H,YOFSET	
9929	B2A1	01	CF	FE LXI B,-YAXLEN-1	
9930	B2A4	CD	0A	B5 CALL VLINE ;DRAW LEFT SIDE	
9931	B2A7	11	AC	02 LXI D,XOFSET+XAXLEN	
9932	B2AA	21	2D	00 LXI H,YOFSET	
9933	B2AD	C3	0A	B5 JMP VLINE ;DRAW RIGHT SIDE	


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 269
=====
9935      B2B0      .      .      .      ;*****
9936      B2B0      .      .      .      ; XAXIS--COMPUTE THE Y COORDINATE OF THE
9937      B2B0      .      .      .      ; HORIZONTAL LINE REPRESENTING THE X AXIS, AND
9938      B2B0      .      .      .      ; DRAW IT IF IT IS ON THE SCREEN.
9939      B2B0      .      .      .      ; THE ONLY TIME THE LINE WILL BE DRAWN IS IF
9940      B2B0      .      .      .      ; YMIN IS NEGATIVE, AND YMAX IS POSITIVE
9941      B2B0      .      .      .      ;*****
9942      B2B0      .      .      .      XAXIS EQU $
9943      B2B0      21  00  00      LXI H,0          ;ASSUME NO AXIS
9944      B2B3      22  F7  FA      SHLD YAX
9945      B2B6      21  B0  FB      LXI H,YMAXBF    ;SEE IF Y MAX IS +
9946      B2B9      CD  6E  BD      CALL LOD
9947      B2BC      CD  59  BD      CALL TST        ;TEST SIGN OF Y MAX
9948      B2BF      C8  .   .      RZ             ;DONE IF ZERO OR -
9949      B2C0      F8  .   .      RM
9950      B2C1      21  B4  FB      LXI H,YMINBF    ;SEE IF YMIN IS -
9951      B2C4      CD  6E  BD      CALL LOD
9952      B2C7      CD  59  BD      CALL TST        ;TEST SIGN
9953      B2CA      F0  .   .      RP             ;DONE IF ZERO OR +
9954      B2CB      .      .      .      ; Y COORDINATE = ABSVAL OF YMIN
9955      B2CB      CD  50  BD      CALL ABS        ;CONVERT TO +
9956      B2CE      21  9E  90      LXI H,YSCALE    ;CONVERT TO SCREEN COORDS
9957      B2D1      CD  8C  BD      CALL MUL
9958      B2D4      D8  .   .      RC             ;EXIT IF OVERFLOW
9959      B2D5      CD  C9  B8      CALL INT        ;HL = Y COORD OF AXIS
9960      B2D8      11  2D  00      LXI D,YOFSET    ;ADD OFFSET FOR FRAME
9961      B2D8      19  .   .      DAD D
9962      B2DC      .      .      .      ; INSURE LINE IS WITHIN FRAME BEFORE DRAWING
9963      B2DC      11  5D  01      LXI D,YOFSET+YAXLEN ;ABOVE TOP?
9964      B2DF      CD  5F  B9      CALL CHKMAX
9965      B2E2      D8  .   .      RC             ;YES, DONT DRAW
9966      B2E3      11  2D  00      LXI D,YOFSET    ;BELOW BOTTOM?
9967      B2E6      CD  55  B9      CALL CHKMIN
9968      B2E9      D8  .   .      RC             ;YES, DONT DRAW
9969      B2EA      22  F7  FA      SHLD YAX        ;STORE Y AXIS COORD
9970      B2ED      .      .      .      ; DRAW HORIZONTAL LINE AT Y = HL ACROSS FRAME
9971      B2ED      11  46  00      LXI D,XOFSET    ;X COORD OF STARTING POINT
9972      B2F0      01  99  FD      LXI B,-XAXLEN-1 ;LENGTH OF LINE
9973      B2F3      C3  06  B5      JMP HLINE       ;DRAW THE LINE
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 270
9975	B2F6	. . .	;*****	
9976	B2F6	. . .	; XTICS--COMPUTE LOCATION OF AND DRAW LABELED AND	
9977	B2F6	. . .	; MINOR TICS ON TOP AND BOTTOM OF FRAME	
9978	B2F6	. . .	; FIRST FIND THE FIRST TIC LESS THAN	
9979	B2F6	. . .	; XMIN. THEN ADD TIC SPACING UNTIL TICS ARE ON	
9980	B2F6	. . .	; THE FRAME (START DRAWING THE TICS).	
9981	B2F6	. . .	; WHEN THE TICS GO OFF OF THE RIGHT SIDE OF THE	
9982	B2F6	. . .	; FRAME, STOP.	
9983	B2F6	. . .	; THIS ROUTINE ASSUMES XMAX > XMIN	
9984	B2F6	. . .	;*****	
9985	B2F6	. . .	XTICS EQU \$	
9986	B2F6	AF . .	XRA A ;CLEAR 'HAVE DRAWN TIC' FLAG	
9987	B2F7	32 E3 FA	STA TICFLG	
9988	B2FA	. . .	; FIRST SEE IF TIC SPACING IS 0. IF SO, DONT	
9989	B2FA	. . .	; DRAW ANY TICS	
9990	B2FA	2A EC FA	LHLD TICPTR ;FETCH TIC SPACING	
9991	B2FD	CD 6E BD	CALL LOD	
9992	B300	CD 59 BD	CALL TST ;IS IT 0??	
9993	B303	C8 . .	RZ ;YES, DONE	
9994	B304	. . .	; COMPUTE ABSOLUTE VALUE OF SPACING	
9995	B304	CD 50 BD	CALL ABS	
9996	B307	2A EC FA	LHLD TICPTR ;STORE ABS VAL	
9997	B30A	CD 3E BD	CALL STR	
9998	B300	. . .	; FIND FIRST TIC TO LEFT OF XMIN.	
9999	B300	. . .	; DIVIDE XMIN BY TIC SPACING TO GET NO. OF TICS	
10000	B300	. . .	; SUBTRACT ONE TO INSURE LESS THAN XMIN	
10001	B300	. . .	; DIVIDE INTO XMIN	
10002	B300	21 8C FB	LXI H,XMINBF ;FETCH MINIMUM VALUE	
10003	B310	CD 6E BD	CALL LOD	
10004	B313	2A EC FA	LHLD TICPTR ;DIVIDE BY TIC SPACING	
10005	B316	CD B4 BD	CALL DIV	
10006	B319	D8 . .	RC	
10007	B31A	. . .	; TRUNCATE FRACTIONAL PART AND SUBTRACT ONE	
10008	B31A	. . .	; THEN MULTIPLY BY TIC SPACING TO GET FIRST	
10009	B31A	. . .	; TIC .LT. XMIN	
10010	B31A	CD 3D B9	CALL TRUNCT ;DELETE FRACTIONAL PART	
10011	B31D	2A EC FA	LHLD TICPTR ;TIC SPACING	
10012	B320	CD 8C BD	CALL MUL ;NO, HAVE FIRST TIC	
10013	B323	D8 . .	RC ;DONE IF OVERFLOW	
10014	B324	21 F3 FA	LXI H,CURTIC ;STORE THE FIRST TIC	
10015	B327	CD 3E BD	CALL STR	
10016	B32A	. . .	; TIC LOOP	
10017	B32A	. . .	; CONVERT CURRENT TIC TO SCREEN COORDINATES	
10018	B32A	. . .	; FP ACCUMULATOR CONTAINS CURRENT TIC	
10020	B32A	. . .	TCX010 EQU \$	
10021	B32A	. . .	; TEST FOR TIC ALMOST ZERO. IF ABS VAL OF TIC IS	
10022	B32A	. . .	; LESS THAN TIC SPACING/2, SET TIC TO 0	
10023	B32A	CD 57 B5	CALL TICZRO	
10024	B32D	. . .	; CONVERT TIC TO SCREEN COORDINATES	

=====					PAGE 271	
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	
=====						
10025	B32D	21	BC	FB	LXI H,XMINBF	;SUBTRACT MIN
10026	B330	CD	D3	BD	CALL SB	
10027	B333	21	A2	90	LXI H,XSCALE	;SCALE
10028	B336	CD	8C	BD	CALL MUL	
10029	B339	D8	.	.	RC	;DONE IF OVERFLOW
10030	B33A	CD	C9	B8	CALL INT	;CONVERT TO INTEGER
10031	B33D	11	46	00	LXI D,XOFSET	;ADD OFFSET FOR FRAME
10032	B340	19	.	.	DAD D	
10033	B341	.	.	.		; HL = X COORD OF TIC
10034	B341	.	.	.		; SEE IF IT IS TO THE RIGHT OF THE FRAME (TOO BIG)
10035	B341	11	AC	02	LXI D,XOFSET+XAXLEN	;MAXIMUM ALLOWED
10036	B344	CD	5F	B9	CALL CHKMAX	;TOO BIG?
10037	B347	D8	.	.	RC	;YES, ALL TICS DRAWN
10038	B348	.	.	.		; TEST FOR TIC WITHIN FRAME (COULD BE TOO FAR
10039	B348	.	.	.		; TO LEFT)
10040	B348	11	46	00	LXI D,XOFSET	;MINIMUM VALUE
10041	B34B	CD	55	B9	CALL CHKMIN	;TIC WITHIN FRAME?
10042	B34E	D2	59	B3	JNC TCX020	;YES
10043	B351	.	.	.		; IF ANY TICS HAVE BEEN DRAWN, AND THIS TIC IS
10044	B351	.	.	.		; TO THE LEFT OF THE FRAME, THEN EXIT (ERROR)
10045	B351	3A	E3	FA	LDA TICFLG	;ANY TICS DRAWN YET?
10046	B354	B7	.	.	ORA A	
10047	B355	CA	70	B3	JZ TCX030	;NO, UPDATE TIC VALUE
10048	B358	C9	.	.	RET	;YES, EXIT
10049	B359	.	.	.	TCX020 EQU \$	
10050	B359	3E	01	.	MVI A,1	;SET 'HAVE DRAWN TIC FLAG'
10051	B35B	32	E3	FA	STA TICFLG	
10052	B35E	.	.	.		; DRAW EITHER LABELED OR UNLABELED TIC
10053	B35E	.	.	.		; HL = X COORD
10054	B35E	EB	.	.	XCHG	;WANT DE = X
10055	B35F	3A	97	FB	LDA APFLG2	;IS TIC LABELED?
10056	B362	E6	80	.	ANI TICLBL	
10057	B364	CA	6D	B3	JZ TCX025	;NO,
10058	B367	CD	97	B3	CALL LABLX	;YES, DRAW LABELED TIC
10059	B36A	C3	70	B3	JMP TCX030	;UPDATE TIC
10060	B36D	.	.	.	TCX025 EQU \$	
10061	B36D	CD	85	B3	CALL MINORX	;DRAW MINOR TIC
10062	B370	.	.	.		; UPDATE CURRENT TIC BY ADDING TIC SPACING
10063	B370	.	.	.	TCX030 EQU \$	
10064	B370	.	.	.		
10065	B370	21	F3	FA	LXI H,CURTIC	;CURRENT TIC
10066	B373	CD	6E	BD	CALL LOD	
10067	B376	2A	EC	FA	LHLD TICPTR	;TIC SPACING
10068	B379	CD	D6	BD	CALL AD	
10069	B37C	21	F3	FA	LXI H,CURTIC	;STRORE IT
10070	B37F	CD	3E	BD	CALL STR	
10071	B382	C3	2A	B3	JMP TCX010	;PROCESS NEW TIC

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 272
10073	B385	. . .	;*****	
10074	B385	. . .	; MINORX--DRAW MINOR X TIC MARK AT TOP AND	
10075	B385	. . .	; BOTTOM OF FRAME	
10076	B385	. . .	; ENTRY DE = X COORDINATE OF TIC	
10077	B385	. . .	; MINLEN = LENTH OF MINOR TIC	
10078	B385	. . .	;*****	
10079	B385	. . .	MINORX EQU \$	
10080	B385	21 2D 00	LXI H,YOFSET ;Y COORD, BOTTOM TIC	
10081	B388	01 FD FF	LXI B,-MINLEN-1 ;TIC LENGTH	
10082	B388	CD 0A B5	CALL VLINE ;DRAW TIC AT BOTTOM	
10083	B38E	21 5B 01	LXI H,YOFSET+YAXLEN-MINLEN ;Y COORD, TOP TI	
10084	B391	CD 0A B5	CALL VLINE ;DRAW TIC AT TOP	
10085	B394	C3 CC B3	JMP LBX010 ;DO TICS ON AXIS	
10086	B397	. . .	;*****	
10087	B397	. . .	; LABLX--DRAW X LABELED X TIC AT TOP AND BOTTOM	
10088	B397	. . .	; OF FRAME, AND ADD LABEL AT BOTTOM	
10089	B397	. . .	; ENTRY DE = X COORD OF TIC	
10090	B397	. . .	; CURTIC = VALUE OF TIC FOR LABEL	
10091	B397	. . .	; LBLEN = LENGTH OF LABELED TIC MARK	
10092	B397	. . .	;*****	
10093	B397	. . .	LABLX EQU \$	
10094	B397	. . .	; DRAW THE LABEL	
10095	B397	D5 . .	PUSH D ;SAVE X COORD	
10096	B398	21 28 00	LXI H,YOFSET-5 ;Y COORD FOR LABEL	
10097	B398	CD 8E B5	CALL GETLBL ;DRAW THE LABEL	
10098	B39E	D1 . .	POP D ;RECALL X COORD	
10099	B39F	21 2D 00	LXI H,YOFSET ;Y COORD, BOTTOM TIC	
10100	B3A2	01 FB FF	LXI B,-LBLEN-1 ;TIC LENGTH	
10101	B3A5	CD 0A B5	CALL VLINE ;DRAW TIC AT BOTTOM	
10102	B3A8	. . .	; IF GRID IS ON, DRAW GRID LINE FROM BOTTOM	
10103	B3A8	. . .	; TO TOP	
10104	B3A8	2A 9A FB	LHLD GRIDBF ;IS GRID ON?	
10105	B3AB	7C . .	MOV A,H ;ANY NON ZERO VALUE MEANS	
10106	B3AC	B5 . .	ORA L ;TO DRAW GRID	
10107	B3AD	CA C6 B3	JZ LBX005 ;NO GRID WANTED	
10108	B3B0	3E 88 .	MVI A,GRDPAT ;LOAD GRID PATTERN	
10109	B3B2	32 C7 FB	STA PAT2	
10110	B3B5	01 CF FE	LXI B,-YAXLEN-1 ;GRID LINE LENGTH	
10111	B3B8	21 2D 00	LXI H,YOFSET ;Y COORD OF GRID LINE	
10112	B3BB	CD 0A B5	CALL VLINE ;DRAW LINE	
10113	B3BE	3E FF .	MVI A,377Q ;SET SOLID PATTERN	
10114	B3C0	32 C7 FB	STA PAT2	
10115	B3C3	01 FB FF	LXI B,-LBLEN-1 ;RESTORE B	
10116	B3C6	. . .	LBX005 EQU \$	
10117	B3C6	21 59 01	LXI H,YOFSET+YAXLEN-LBLEN ;Y COORD TOP TIC	
10118	B3C9	CD 0A B5	CALL VLINE ;DRAW TIC AT TOP	
10119	B3CC	. . .	; DRAW TIC ON AXIS, IF AXIS PRESENT	
10120	B3CC	. . .	LBX010 EQU \$	
10121	B3CC	2A F7 FA	LHLD YAX ;IS THERE AN AXIS?	
10122	B3CF	7C . .	MOV A,H	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 273
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 273
10123	B3D0	B5	.	ORA L	;NO AXIS IF 0
10124	B3D1	C8	.	RZ	;NO AXIS, DONT DRAW TIC
10125	B3D2	E5	.	PUSH H	;SAVE STARTING POINT
10126	B3D3	09	.	DAD B	;SUBTRACT TIC LENGTH
10127	B3D4	23	.	INX H	;LENGTH OFF BY ONE
10128	B3D5	CD	0A B5	CALL VLINE	;DRAW THE TIC
10129	B3D8	E1	.	POP H	;(RECALL STARTING POINT)
10130	B3D9	C3	0A B5	JMP VLINE	;ON BOTH SIDES OF AXIS

```
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 274
10132	B3DC	. . .	;*****	
10133	B3DC	. . .	; YAXIS--COMPUTE THE X COORDINATE OF THE	
10134	B3DC	. . .	; VERTICAL LINE REPRESENTING THE Y AXIS, AND	
10135	B3DC	. . .	; DRAW IT IF IT IS ON THE SCREEN.	
10136	B3DC	. . .	; THE ONLY TIME THE LINE WILL BE DRAWN IS IF	
10137	B3DC	. . .	; XMIN IS NEGATIVE, AND XMAX IS POSITIVE	
10138	B3DC	. . .	;*****	
10139	B3DC	. . .	YAXIS EQU \$	
10140	B3DC	21 00 00	LXI H,0 ;ASSUME NO AXIS	
10141	B3DF	22 F9 FA	SHLD XAX	
10142	B3E2	21 B8 FB	LXI H,XMAXBF ;SEE IF X MAX IS +	
10143	B3E5	CD 6E BD	CALL LOD	
10144	B3E8	CD 59 BD	CALL TST ;TEST SIGN OF X MAX	
10145	B3EB	C8 . .	RZ ;DONE IF ZERO OR -	
10146	B3EC	F8 . .	RM	
10147	B3ED	21 BC FB	LXI H,XMINBF ;SEE IF XMIN IS -	
10148	B3F0	CD 6E BD	CALL LOD	
10149	B3F3	CD 59 BD	CALL TST ;TEST SIGN	
10150	B3F6	F0 . .	RP ;DONE IF ZERO OR +	
10151	B3F7	. . .	; X COORDINATE = ABSVAL OF XMIN	
10152	B3F7	CD 50 BD	CALL ABS ;CONVERT TO +	
10153	B3FA	21 A2 90	LXI H,XSCALE ;CONVERT TO SCREEN COORDS	
10154	B3FD	CD 8C BD	CALL MUL	
10155	B400	D8 . .	RC ;DONE IF OVERFLOW	
10156	B401	CD C9 B8	CALL INT ;HL = Y COORD OF AXIS	
10157	B404	11 46 00	LXI D,XOFSET ;ADD OFFSET FOR FRAME	
10158	B407	19 . .	DAD D	
10159	B408	. . .	; INSURE LINE IS WITHIN FRAME BEFORE DRAWING	
10160	B408	11 AC 02	LXI D,XOFSET+XAXLEN ;TOO FAR TO RIGHT?	
10161	B40B	CD 5F B9	CALL CHKMAX	
10162	B40E	D8 . .	RC ;YES, DONT DRAW	
10163	B40F	11 46 00	LXI D,XOFSET ;TOO FAR TO LEFT?	
10164	B412	CD 55 B9	CALL CHKMIN	
10165	B415	D8 . .	RC ;YES, DONT DRAW	
10166	B416	22 F9 FA	SHLD XAX ;STORE AXIS LOCATION	
10167	B419	. . .	; DRAW VERTICAL LINE AT X = HL ACROSS FRAME	
10168	B419	11 2D 00	LXI D,YOFSET ;Y COORD OF STARTING POINT	
10169	B41C	01 CF FE	LXI B,-YAXLEN-1 ;LENGTH OF LINE	
10170	B41F	EB . .	XCHG ;DE=X, HL=Y	
10171	B420	C3 0A B5	JMP VLINE ;DRAW THE LINE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 275
=====
10173    B423      . . .      ;*****
10174    B423      . . .      ; YTICS--COMPUTE LOCATION OF AND DRAW LABELED AND
10175    B423      . . .      ; MINOR TICS ON RIGHT AND LEFT SIDES OF FRAME.
10176    B423      . . .      ; FIRST FIND THE FIRST TIC LESS THAN
10177    B423      . . .      ; YMIN. THEN ADD TIC SPACING UNTIL TICS ARE ON
10178    B423      . . .      ; THE FRAME (START DRAWING THE TICS).
10179    B423      . . .      ; WHEN THE TICS GO OFF OF THE RIGHT SIDE OF THE
10180    B423      . . .      ; FRAME, STOP.
10181    B423      . . .      ; THIS ROUTINE ASSUMES YMAX > YMIN
10182    B423      . . .      ;*****
10183    B423      . . .      YTICS EQU $
10184    B423      AF . . .      XRA A ;CLEAR 'HAVE DRAWN TIC' FLAG
10185    B424      32 E3 FA      STA TICFLG
10186    B427      . . .      ; FIRST SEE IF TIC SPACING IS 0. IF SO, DONT
10187    B427      . . .      ; DRAW ANY TICS
10188    B427      2A EC FA      LHLD TICPTR ;FETCH TIC SPACING
10189    B42A      CD 6E BD      CALL LOD
10190    B42D      CD 59 BD      CALL TST ;IS IT 0??
10191    B430      C8 . . .      RZ ;YES, DONE
10192    B431      . . .      ; COMPUTE ABSOLUTE VALUE OF SPACING
10193    B431      CD 50 BD      CALL ABS
10194    B434      2A EC FA      LHLD TICPTR ;STORE ABS VAL
10195    B437      CD 3E BD      CALL STR
10196    B43A      . . .      ; FIND FIRST TIC TO LEFT OF MIN VALUE
10197    B43A      . . .      ; DIVIDE Y MIN BY TIC SPACING TO GET NO. OF TICS
10198    B43A      . . .      ; AND SUBTRACT ONE TO INSURE .LT. MIN
10199    B43A      . . .      ; DIVIDE INTO YMIN
10200    B43A      21 B4 FB      LXI H,YMINBF ;FETCH MINIMUM VALUE
10201    B43D      CD 6E BD      CALL LOD
10202    B440      2A EC FA      LHLD TICPTR ;DIVIDE BY TIC SPACING
10203    B443      CD B4 BD      CALL DIV
10204    B446      D8 . . .      RC
10205    B447      . . .      ; TRUNCATE FRACTIONAL PART AND SUBTRACT ONE
10206    B447      . . .      ; THIS VALUE WHEN MULTIPLIED BY TIC SPACING
10207    B447      . . .      ; GIVES FIRST TIC .LT. YMIN
10208    B447      CD 3D B9      CALL TRUNCT ;DELETE FRACTIONAL PART
10209    B44A      2A EC FA      LHLD TICPTR ;TIC SPACING
10210    B44D      CD 8C BD      CALL MUL ;NOW HAVE FIRST TIC
10211    B450      D8 . . .      RC ;EXIT IF OVERFLOW
10212    B451      21 F3 FA      LXI H,CURTIC ;STORE THE FIRST TIC
10213    B454      CD 3E BD      CALL STR
10214    B457      . . .      ;
10215    B457      . . .      ; TIC LOOP
10216    B457      . . .      ; CONVERT CURRENT TIC TO SCREEN COORDINATES
10217    B457      . . .      ; FP ACCUMULATOR CONTAINS CURRENT TIC
10218    B457      . . .      TCY010 EQU $
10219    B457      . . .      ; IF ABS VAL OF TIC IS .LT. LABEL SPACING/2,
10220    B457      . . .      ; SET TIC TO ZERO
10221    B457      CD 57 B5      CALL TICZRO ;TEST FOR ALMOST ZERO TIC
10222    B45A      . . .      ; CONVERT TIC TO SCREEN COORDINATES
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 276
10223	B45A	21	B4	FB	LXI H,YMINBF ;SUBTRACT MIN	
10224	B45D	CD	D3	BD	CALL S8	
10225	B460	21	9E	90	LXI H,YSCALE ;SCALE	
10226	B463	CD	8C	BD	CALL MUL	
10227	B466	08	.	.	RC ;EXIT IF OVERFLOW	
10228	B467	CD	C9	B8	CALL INT ;CONVERT TO INTEGER	
10229	B46A	11	2D	00	LXI D,YOFSET ;ADD OFFSET FOR FRAME	
10230	B46D	19	.	.	DAD D	
10231	B46E	.	.	.	; HL = Y COORD OF TIC	
10232	B46E	.	.	.	; SEE IF IT IS ABOVE THE FRAME (TOO BIG)	
10233	B46E	11	5D	01	LXI D,YOFSET+YAXLEN ;MAXIMUM ALLOWED	
10234	B471	CD	5F	B9	CALL CHKMAX ;TOO BIG?	
10235	B474	08	.	.	RC ;YES, ALL TICS DRAWN	
10236	B475	.	.	.	; TEST FOR TIC WITHIN FRAME (COULD BE BELOW	
10237	B475	.	.	.	; THE FRAME)	
10238	B475	11	2D	00	LXI D,YOFSET ;MINIMUM VALUE	
10239	B478	CD	55	B9	CALL CHKMIN ;TIC WITHIN FRAME?	
10240	B47B	D2	86	B4	JNC TCY020 ;YES	
10241	B47E	.	.	.	; IF ANY TICS HAVE BEEN DRAWN, AND THIS TIC IS TOO	
10242	B47E	.	.	.	; FAR TO LEFT, EXIT (ERROR)	
10243	B47E	3A	E3	FA	LDA TICFLG ;ANY TICS DRAWN?	
10244	B481	B7	.	.	ORA A	
10245	B482	CA	9C	B4	JZ TCY030 ;NO, UPDATE TIC VALUE	
10246	B485	C9	.	.	RET ;YES, EXIT	
10247	B486	.	.	.	TCY020 EQU \$	
10248	B486	3E	01	.	MVI A,1 ;SET 'HAVE DRAWN TIC FLAG'	
10249	B488	32	E3	FA	STA TICFLG	
10250	B48B	.	.	.	; DRAW EITHER LABELED OR UNLABELED TIC	
10251	B48B	.	.	.	; HL = Y COORD	
10252	B48B	3A	97	FB	LDA APFLG2 ;LABELED?	
10253	B48E	E6	80	.	ANI TICLBL	
10254	B490	CA	99	B4	JZ TCY025 ;NO	
10255	B493	CD	C3	B4	CALL LABLY ;YES, DRAW LABELED TIC	
10256	B496	C3	9C	B4	JMP TCY030	
10257	B499	.	.	.	TCY025 EQU \$	
10258	B499	CD	B1	B4	CALL MINORY ;DRAW UNLABELED TIC	
10259	B49C	.	.	.	; UPDATE CURRENT TIC BY ADDING TIC SPACING	
10260	B49C	.	.	.	TCY030 EQU \$	
10261	B49C	21	F3	FA	LXI H,CURTIC ;CURRENT TIC	
10262	B49F	CD	6E	BD	CALL LOD	
10263	B4A2	2A	EC	FA	LHLD TICPTR ;TIC SPACING	
10264	B4A5	CD	D6	BD	CALL AD	
10265	B4A8	21	F3	FA	LXI H,CURTIC ;STRORE IT	
10266	B4AB	CD	3E	BD	CALL STR	
10267	B4AE	C3	57	B4	JMP TCY010 ;PROCESS NEW TIC	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS
10269	B4B1	. . .	;*****
10270	B4B1	. . .	; MINORY--DRAW MINOR Y TIC MARK AT RIGHT AND
10271	B4B1	. . .	; LEFT SIDES OF FRAME
10272	B4B1	. . .	; ENTRY HL = Y COORD OF TIC
10273	B4B1	. . .	; MINLEN = LENTH OF MINOR TIC
10274	B4B1	. . .	;*****
10275	B4B1	. . .	MINORY EQU \$
10276	B4B1	11 46 00	LXI D,XOFSET ;X COORD, RIGHT TIC
10277	B4B4	01 FD FF	LXI B,-MINLEN-1 ;TIC LENGTH
10278	B4B7	CD 06 B5	CALL HLINE ;DRAW TIC AT LEFT SIDE
10279	B4BA	11 AA 02	LXI D,XOFSET+XAXLEN-MINLEN ;X TIC COORD
10280	B4BD	CD 06 B5	CALL HLINE ;DRAW TIC AT RIGHT SIDE
10281	B4C0	C3 F7 B4	JMP LBY010 ;DRAW TIC ON AXIS
10282	B4C3	. . .	;*****
10283	B4C3	. . .	; LABLY--DRAW X LABELED Y TIC AT RIGHT AND LEFT
10284	B4C3	. . .	; SIDES OF FRAME, AND ADD LABEL AT LEFT
10285	B4C3	. . .	; ENTRY HL = Y COORD OF TIC
10286	B4C3	. . .	; CURTIC = VALUE OF TIC FOR LABEL
10287	B4C3	. . .	; LBLEN = LENGTH OF LABELED TIC MARK
10288	B4C3	. . .	;*****
10289	B4C3	. . .	LABLY EQU \$
10290	B4C3	E5 . .	PUSH H ;SAVE Y COORD
10291	B4C4	11 41 00	LXI D,XOFSET-5 ;X COORD, RIGHT TIC
10292	B4C7	CD 8E B5	CALL GETLBL ;DRAW THE LABEL
10293	B4CA	E1 . .	POP H ;RESTORE Y COORD
10294	B4CB	11 46 00	LXI D,XOFSET ;X COORD, RIGHT TIC
10295	B4CE	01 FB FF	LXI B,-LBLEN-1 ;TIC LENGTH
10296	B4D1	CD 06 B5	CALL HLINE ;DRAW TIC AT LEFT SIDE
10297	B4D4	. . .	; IF GRID IS WANTED, DRAW GRID FROM RIGHT TO LEFT
10298	B4D4	E5 . .	PUSH H ;SAVE H = Y COORD
10299	B4D5	2A 9A FB	LHLD GRIDBF ;ANY NON ZERO ENTRY MEANS
10300	B4D8	7C . .	MOV A,H ;TO DRAW THE GRIDLINE
10301	B4D9	B5 . .	ORA L
10302	B4DA	E1 . .	POP H ;(RESTORE H)
10303	B4DB	CA F1 B4	JZ LBY005 ;NO GRID WANTED
10304	B4DE	3E 88 .	MVI A,GRDPAT ;LOAD GRID PATTERN
10305	B4E0	32 C7 FB	STA PAT2
10306	B4E3	01 99 FD	LXI B,-XAXLEN-1 ;GRID LENGTH
10307	B4E6	CD 06 B5	CALL HLINE ;DRAW THE LINE
10308	B4E9	3E FF .	MVI A,377Q ;RESTORE SOLID PATTERN
10309	B4EB	32 C7 FB	STA PAT2
10310	B4EE	01 FB FF	LXI B,-LBLEN-1 ;RESTORE B
10311	B4F1	. . .	LBY005 EQU \$
10312	B4F1	11 A8 02	LXI D,XOFSET+XAXLEN-LBLEN ;X TIC COORD
10313	B4F4	CD 06 B5	CALL HLINE ;DRAW TIC AT RIGHT SIDE
10314	B4F7	. . .	; DRAW TIC ON AXIS, IF AXIS PRESENT
10315	B4F7	. . .	LBY010 EQU \$
10316	B4F7	EB . .	XCHG
10317	B4F8	2A F9 FA	LHLD XAX ;IS THERE AN AXIS?
10318	B4FB	7C . .	MOV A,H

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 278
=====
10319    B4FC    B5 . .      ORA L
10320    B4FD    C8 . .      RZ ;NO, DONT DRAW TIC
10321    B4FE    E5 . .      PUSH H ;SAVE STARTING POINT
10322    B4FF    09 . .      DAD B ;SUBTRACT TIC LENGTH
10323    B500    23 . .      INX H ;OFF BY ONE
10324    B501    EB . .      XCHG
10325    B502    CD 06 B5    CALL HLINE ;DRAW THE TIC
10326    B505    D1 . .      POP D ;(RECALL STARTING POINT)
10327    B506    . . .      ; FALL INTO HLINE TO DRAW TIC ON OTHER SIDE
=====
```

				PAGE 279	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	
10329	B506	.	.	.	;*****
10330	B506	.	.	.	; HLINE--DRAW HORIZONTAL LINE FROM LEFT TO RIGHT
10331	B506	.	.	.	; ENTRY DE = X COORD OF START
10332	B506	.	.	.	; HL = Y COORD OF START
10333	B506	.	.	.	; BC = -(LENGTH)
10334	B506	.	.	.	; EXIT A DESTROYED
10335	B506	.	.	.	;*****
10336	B506	.	.	.	HLINE EQU \$
10337	B506	AF	.	.	XRA A ;SET HORIZONTAL LINE FLAG
10338	B507	C3	0C	B5	JMP LINE1
10339	B50A	.	.	.	;*****
10340	B50A	.	.	.	; VLINE--DRAW VERTICAL LINE FROM BOTTOM TO TOP
10341	B50A	.	.	.	; REGISTERS SAME AS HLINE
10342	B50A	.	.	.	;*****
10343	B50A	.	.	.	VLINE EQU \$
10344	B50A	3E	FF	.	MVI A,377Q ;SET VERTICAL LINE FLAG
10345	B50C	.	.	.	LINE1 EQU \$
10346	B50C	C5	.	.	PUSH B ;SAVE REGISTERS
10347	B50D	D5	.	.	PUSH D
10348	B50E	E5	.	.	PUSH H
10349	B50F	B7	.	.	ORA A ;SET FLAGS
10350	B510	F5	.	.	PUSH PSW
10351	B511	D5	.	.	PUSH D ;SAVE X COORD
10352	B512	CD	5B	67	CALL MPY45 ;COMPUTE 45*Y
10353	B515	D1	.	.	POP D ;RECALL X
10354	B516	CD	6F	67	CALL GETWA ;COMPUTE 18 BIT ADDRESS
10355	B519	CD	87	A2	CALL WAIT ;WAIT FOR IDLE HW
10356	B51C	22	0E	89	SHLD LSBWA ;SEND LOWER 12 BITS
10357	B51F	32	0C	89	STA MSBWA ;SEND UPPER 12 BITS]
10358	B522	.	.	.	; SEND DOT COUNT
10359	B522	69	.	.	MOV L,C
10360	B523	60	.	.	MOV H,B
10361	B524	22	12	89	SHLD DC ;SEND DOT COUNT
10362	B527	.	.	.	; SEND D1 AND M1 FOR DIRECTION
10363	B527	21	01	00	LXI H,1 ;ASSUME HORIZONTAL
10364	B52A	F1	.	.	POP PSW ;RECALL DIRECTION
10365	B52B	CA	31	B5	JZ LIN010 ;IT IS
10366	B52E	21	30	FD	LXI H,-720 ;NO, ITS VERTICAL
10367	B531	.	.	.	LIN010 EQU \$
10368	B531	22	1A	89	SHLD M1 ;SEND INCREMENT
10369	B534	32	18	89	STA SIGNM1
10370	B537	CD	0F	A4	CALL VSETUP ;SET CONSTANT PARAMETERS
10371	B53A	32	21	89	STA SCALER ;SET PRESCALE TO 0
10372	B53D	.	.	.	; SET DRAWING MODE
10373	B53D	3A	B5	90	LDA CURMOD
10374	B540	F6	04	.	ORI PATENB ;ALLOW PATTERN
10375	B542	32	41	89	STA HCEJK
10376	B545	3A	C7	FB	LDA PAT2
10377	B548	32	40	89	STA PATERN
10378	B54B	.	.	.	; CONTROLLER LOADED, SO DRAW THE LINE

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 280
10379	B54B	CD	22	67	CALL HWGO ;START HW GOING	
10380	B54E	3E	08	.	MVI A,NEWWA ;MSUT RECOMPUTE WA	
10381	B550	CD	26	A2	CALL STFLG1	
10382	B553	E1	.	.	POP H ;RESTORE REGISTERS	
10383	B554	D1	.	.	POP D	
10384	B555	C1	.	.	POP B	
10385	B556	C9	.	.	RET	
10386	B557	.	.	.	*****	
10387	B557	.	.	.	; TICZRO--TEST FOR TIC ALMOST ZERO	
10388	B557	.	.	.	; IF ABS VAL OF TIC IS LESS THAN ABS VAL OF	
10389	B557	.	.	.	; TIC SPACING/2 , IT IS SET TO ZERO	
10390	B557	.	.	.	; ENTRY & EXIT -- FPACCUM CONTAINS TIC	
10391	B557	.	.	.	*****	
10392	B557	.	.	.	TICZRO EQU \$	
10393	B557	21	E8	FA	LXI H,FPSAVE ;SAVE TIC	
10394	B55A	CD	3E	BD	CALL STR	
10395	B55D	CD	50	BD	CALL ABS ;WANT ABS VALUE	
10396	B560	21	E4	FA	LXI H,FPSAV2 ;SAVE ABS VAL	
10397	B563	CD	3E	BD	CALL STR	
10398	B566	2A	EC	FA	LHLD TICPTR ;LOAD TIC SPACING	
10399	B569	CD	6E	BD	CALL LOD	
10400	B56C	21	8A	B5	LXI H,FPTWO ;DIVIDE BY TWO	
10401	B56F	CD	84	BD	CALL DIV	
10402	B572	21	E4	FA	LXI H,FPSAV2 ;SUBTRACT TIC	
10403	B575	CD	D3	BD	CALL SB	
10404	B578	.	.	.	; IF RESULT IS -, TIC IS OK	
10405	B578	.	.	.	; IF RESULT IS +, TIC WILL BE ASSUMED TO BE 0	
10406	B578	F5	.	.	PUSH PSW ;SAVE FLAGS	
10407	B579	21	E8	FA	LXI H,FPSAVE ;RESTORE TIC	
10408	B57C	CD	6E	BD	CALL LOD	
10409	B57F	F1	.	.	POP PSW ;GET FLAGS	
10410	B580	F8	.	.	RM ;OK IF WAS -	
10411	B581	CD	46	BD	CALL ZRO ;SET TIC TO ZERO	
10412	B584	21	F3	FA	LXI H,CURTIC ;STORE ZEROED CURRENT TIC	
10413	B587	C3	3E	BD	JMP STR	
10414	B58A	.	.	.	;	
10415	B58A	82	00	00	FPTWO DB 202Q,000Q,000Q,000Q ;2 IN FLOAT POINT	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
10417	B58E	.	.	;*****
10418	B58E	.	.	; GETLBL--DRAW LABEL FOR AUTOPLT
10419	B58E	.	.	; ENTRY DE = X COORD
10420	B58E	.	.	; HL = Y COORD
10421	B58E	.	.	; CURTIC = VALUE OF TIC
10422	B58E	.	.	;*****
10423	B58E	.	.	GETLBL EQU \$
10424	B58E	22	DC 90	SHLD YCURR ;SAVE POINT FOR SNDLBL
10425	B591	EB	.	XCHG
10426	B592	22	DE 90	SHLD XCURR
10427	B595	21	F3 FA	LXI H,CURTIC ;LOAD TIC VALUE
10428	B598	CD	6E BD	CALL LOD
10429	B59B	21	35 FB	LXI H,NUMBUF+40 ;WHERE ASCII LABEL WILL G
10430	B59E	CD	16 BC	CALL OU ;CONVERT TIC TO ASCII
10431	B5A1	.	.	;*****
10432	B5A1	.	.	; ROUND OFF VALUE TO PROPER NUMBER OF SIGNIFICANT
10433	B5A1	.	.	; DIGITS BY DOING ASCII ADD
10434	B5A1	.	.	; ALSO, IF EXPONENTIAL FORMAT, DO POSSIBLE CONVER
10435	B5A1	.	.	; SION BACK TO REGULAR FORMAT
10436	B5A1	.	.	;*****
10437	B5A1	CD	B5 B6	CALL LBLRND
10438	B5A4	.	.	;*****
10439	B5A4	.	.	; GO THRU LABEL AND ADD 60B TO COMPENSATE FOR
10440	B5A4	.	.	; INTEL SOFTWARE, AND DELETE UNWANTED CHARACTERS
10441	B5A4	.	.	; LBLFMT HAS COMPUTED HOW MANY
10442	B5A4	.	.	; CHARACTERS AFTER THE DECIMAL POINT ARE TO BE
10443	B5A4	.	.	; DISPLAYED, AND WHETHER A DECIMAL POINT IS TO
10444	B5A4	.	.	; BE DRAWN OR NOT. THE LABEL IN NUMBUF IS
10445	B5A4	.	.	; GUARENTEED TO HAVE A DECIMAL POINT, AND ENDS
10446	B5A4	.	.	; WITH EITHER 4 BLANKS, OR AN EXP FIELD.
10447	B5A4	.	.	;*****
10448	B5A4	AF	.	XRA A ;RESET E LOCATION
10449	B5A5	32	EE FA	STA ECNT
10450	B5A8	4F	.	MOV C,A ;C = CHAR INDEX
10451	B5A9	21	EF FA	LXI H,CHRCNT ;NO. OF DIGITS AFTER DEC PT
10452	B5AC	46	.	MOV B,M
10453	B5AD	11	0D FB	LXI D,NUMBUF ;DESTINATION OF FMTED LABEL
10454	B5B0	2A	F0 FA	LHLD LBLPTR ;WHERE LABEL IS NOW
10455	B5B3	7E	.	MOV A,M ;FETCH FIRST CHAR
10456	B5B4	FE	F0 .	CPI 360Q ;IS IT A SPACE?
10457	B5B6	C2	BA B5	JNZ GLB010 ;NO, LEAVE AS IS
10458	B5B9	23	.	INX H ;IGNORE LEADING SPACE
10459	B5BA	.	.	GLB010 EQU \$
10460	B5BA	.	.	; GO THRU LOOKING FOR DEC POINT, CONVERTING AS
10461	B5BA	.	.	; CHARS ARE READ
10462	B5BA	7E	.	MOV A,M ;FETCH CHAR
10463	B5BB	C6	30 .	ADI 60Q ;CONVERT TO ASCII
10464	B5BD	FE	2E .	CPI 56Q ;IS IT A DEC POINT?
10465	B5BF	CA	C9 B5	JZ GLB020 ;YES, EXIT THIS LOOP
10466	B5C2	12	.	STAX D ;NO, JUST STORE THE CHAR

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 282
10467	B5C3	23 . .	INX H ;UPDATE POINTERS	
10468	B5C4	13 . .	INX D	
10469	B5C5	0C . .	INR C ;UPDATE INDEX	
10470	B5C6	C3 BA B5	JMP GLB010 ;DO NEXT CHAR	
10471	B5C9	. . .	GLB020 EQU \$	
10472	B5C9	. . .	; HAVE DECIMAL POINT, DONT STORE IT IF INTEGER	
10473	B5C9	. . .	; FORMAT	
10474	B5C9	3A 97 FB	LDA APFLG2 ;WANT DECIMAL POINT?	
10475	B5CC	E6 20 .	ANI DECPNT	
10476	B5CE	CA D6 B5	JZ GLB030 ;NO, DONT STORE .	
10477	B5D1	3E 2E .	MVI A,56Q ;YES, STORE A DEC POINT	
10478	B5D3	12 . .	STAX D	
10479	B5D4	13 . .	INX D ;UPDATE POINTER	
10480	B5D5	0C . .	INR C ;UPDATE INDEX	
10481	B5D6	. . .	GLB030 EQU \$	
10482	B5D6	23 . .	INX H ;GET NEXT CHAR	
10483	B5D7	. . .	GLB040 EQU \$	
10484	B5D7	. . .	; NOW LOOK FOR A SPACE OR AN E	
10485	B5D7	. . .	; DONT STORE DIGITS AFTER FORMAT COUNT IS	
10486	B5D7	. . .	; REACHED	
10487	B5D7	7E . .	MOV A,M ;FETCH CHAR	
10488	B5D8	FE F0 .	CPI 360Q ;SPACE?	
10489	B5DA	CA 02 B6	JZ GLB070 ;YES, DONE C=LABEL LENGTH	
10490	B5DD	C6 30 .	ADI 60Q ;NO, CONVERT TO ASCII	
10491	B5DF	FE 45 .	CPI 105Q ;CAP E?	
10492	B5E1	CA EF B5	JZ GLB050 ;YES, PROCESS EXP FIELD	
10493	B5E4	23 . .	INX H ;UPDATE SOURCE POINTER	
10494	B5E5	05 . .	DCR B ;UPDATE #CHAR AFTER DEC PT	
10495	B5E6	FA D7 B5	JM GLB040 ;DONT STORE IF LIMIT REACHED	
10496	B5E9	12 . .	STAX D ;STORE CHAR	
10497	B5EA	13 . .	INX D ;UPDATE DESTINATION	
10498	B5EB	0C . .	INR C ;UPDATE LABEL LENGTH	
10499	B5EC	C3 D7 B5	JMP GLB040	
10500	B5EF	. . .	GLB050 EQU \$	
10501	B5EF	. . .	; HAVE E, PROCESS LAST 3 CHAR OF EXP FLD	
10502	B5EF	12 . .	STAX D ;STORE THE E	
10503	B5F0	0C . .	INR C ;UPDATE LENGTH	
10504	B5F1	79 . .	MOV A,C ;STORE LOCATION OF E	
10505	B5F2	32 EE FA	STA ECNT	
10506	B5F5	06 03 .	MVI B,3 ;CNTR FOR LAST 3 CHAR	
10507	B5F7	. . .	GLB060 EQU \$	
10508	B5F7	23 . .	INX H ;UPDATE SOURCE POINTER	
10509	B5F8	7E . .	MOV A,M ;FETCH CHAR	
10510	B5F9	C6 30 .	ADI 60Q ;CONVERT TO ASCII	
10511	B5FB	13 . .	INX D ;UPDATE DESTINATION	
10512	B5FC	12 . .	STAX D ;STORE CHAR	
10513	B5FD	0C . .	INR C ;UPDATE LENGTH	
10514	B5FE	05 . .	DCR B ;ALL 3 DONE?	
10515	B5FF	C2 F7 B5	JNZ GLB060 ;NO	
10516	B602	. . .	GLB070 EQU \$	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 283
10517	B602	.	.	.	; NUMBUF = NEW LABEL
10518	B602	.	.	.	; C = LENGTH OF LABEL
10519	B602	.	.	.	; IF LABEL IS 9 CHAR OR LESS, IT WILL ALL FIT ON
10520	B602	.	.	.	; ONE LINE. LONG IS ALREADY SET TO PROPER VALUE
10521	B602	79	.	.	MOV A,C ;SET LENGTH
10522	B603	FE	0A	.	CPI 10 ;9 OR LESS?
10523	B605	D2	0E	B6	JNC GLB080 ;NO, TOO BIG
10524	B608	.	.	.	; DRAW SHORT LABEL
10525	B608	21	0D	FB	LXI H,NUMBUF
10526	B60B	C3	28	99	JMP SNDLBL
10527	B60E	.	.	.	GLB080 EQU \$
10528	B60E	.	.	.	; BREAK OUT EXPONENTIAL PART AND DRAW SEPARATELY
10529	B60E	.	.	.	; DRAW MANTISSA FIRST
10530	B60E	3A	EE	FA	LDA ECNT ;WHERE THE EXP FIELD STARTS
10531	B611	3D	.	.	DCR A ;NO. OF CHARS BEFORE THE E
10532	B612	21	0D	FB	LXI H,NUMBUF
10533	B615	CD	28	99	CALL SNDLBL ;DRAW IT
10534	B618	.	.	.	; IF X LABEL (IN CENTER MODE) MOVE DOWN ONE
10535	B618	.	.	.	; LINE AND DRAW CENTERED
10536	B618	3A	97	90	LDA GFLGS6 ;IN CENTERED MODE?
10537	B61B	E6	08	.	ANI CNTR
10538	B61D	CA	3B	B6	JZ GLB100 ;NO, PROCESS Y EXPONENT
10539	B620	.	.	.	GLB090 EQU \$
10540	B620	.	.	.	; MOVE DOWN ONE LINE
10541	B620	2A	83	90	LHLD YLFINC
10542	B623	EB	.	.	XCHG
10543	B624	2A	DC	90	LHLD YCURR
10544	B627	19	.	.	DAD D
10545	B628	22	DC	90	SHLD YCURR
10546	B62B	.	.	.	; DRAW NUMBUF STARTING AT EXP
10547	B62B	21	0D	FB	LXI H,NUMBUF
10548	B62E	3A	EE	FA	LDA ECNT ;WHERE THE E IS
10549	B631	3D	.	.	DCR A
10550	B632	5F	.	.	MOV E,A ;ADD LENGTH TO BASE
10551	B633	16	00	.	MVI D,0
10552	B635	19	.	.	DAD D ;HL = PTR TO EXP FIELD
10553	B636	3E	04	.	MVI A,4 ;4 CHAR FOR SURE
10554	B638	C3	28	99	JMP SNDLBL ;DRAW THE EXP FIELD
10555	B63B	.	.	.	GLB100 EQU \$
10556	B63B	.	.	.	; Y EXP. FIELD
10557	B63B	.	.	.	; MOVE X COORD TO WHERE FIRST CHARACTER OF MANTISS
10558	B63B	.	.	.	; A IS
10559	B63B	3A	EE	FA	LDA ECNT ;WHERE THE E IS
10560	B63E	3D	.	.	DCR A ;NO. OF CHARS IN LINE
10561	B63F	3D	.	.	DCR A
10562	B640	2A	89	90	LHLD XCHINC ;WANT TO BACKSPACE
10563	B643	CD	31	78	CALL MPY1 ;THAT MANY CHARACTERS
10564	B646	CD	09	A3	CALL NEGATE ;BACKSPACE INC
10565	B649	EB	.	.	XCHG
10566	B64A	2A	DE	90	LHLD XCURR

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 284
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 284
10567	B64D	19	. .	DAD D ;BACK AT START OF LINE	
10568	B64E	22	DE 90	SHLD XCURR	
10569	B651	.	. .	; CHANGE LORG TO LEFT JUSTIFY	
10570	B651	3A	97 90	LDA GFLGS6	
10571	B654	F5	. .	PUSH PSW ;SAVE CURRENT LORG	
10572	B655	E6	E7 .	ANI -1-RTJUST-CNTR	
10573	B657	32	97 90	STA GFLGS6	
10574	B65A	.	. .	; DO A LINE FEED, AND DRAW EXP FIELD	
10575	B65A	CD	20 B6	CALL GLB090	
10576	B65D	F1	. .	POP PSW ;RESTORE LORG	
10577	B65E	32	97 90	STA GFLGS6	
10578	B661	C9	. .	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 285
10580	B662	.	.	. ;*****	
10581	B662	.	.	. ; LBLFMT--EXAMINE TIC SPACING FIELD IN MENU	
10582	B662	.	.	. ; TO DETERMINE HOW MANY CHAR AFTER DECIMAL	
10583	B662	.	.	. ; POINT TO PUT IN LABEL. IF NO DEC POINT, THEN	
10584	B662	.	.	. ; SPECIFY INTEGER LABELS. THE TIC VALUE WILL	
10585	B662	.	.	. ; BE ROUNDED OFF TO THE SAME NUMBER OF PLACES	
10586	B662	.	.	. ; AS USED IN THE TIC SPACING FIELD	
10587	B662	.	.	. ; ENTRY HL = POINTER TO X OR Y TIC FIELD	
10588	B662	.	.	. ; EXIT CHRCNT = NO. OF CHAR AFTER DEC POINT	
10589	B662	.	.	. ; DECPNT SET IF DECIMAL POINT WANTED IN TIC	
10590	B662	.	.	. ;*****	
10591	B662	.	.	. LBLFMT EQU \$	
10592	B662	CD	DF	AE CALL TRIM ;DELETE LEADING BLANKS	
10593	B665	01	00	00 LXI B,0 ;RESET COUNT, DEC. POINT FLA	
10594	B668	.	.	. ; TEST FOR INITIAL SIGN	
10595	B668	7E	.	. MOV A,M ;IGNORE LEADING + OR -	
10596	B669	FE	2B	. CPI 53Q ;+ ??	
10597	B668	CA	82	B6 JZ LBF025 ;YES, IGNORE	
10598	B66E	FE	2D	. CPI 55Q ;- ??	
10599	B670	CA	82	B6 JZ LBF025 ;YES, IGNORE	
10600	B673	.	.	. ; LOOP UNTIL DECIMAL POINT OR NON 0-9 IS FOUND	
10601	B673	.	.	. LBF010 EQU \$	
10602	B673	FE	2E	. CPI 56Q ;DECIMAL POINT?	
10603	B675	CA	87	B6 JZ LBF030 ;YES, EXIT	
10604	B678	FE	30	. CPI 60Q ;ZERO?	
10605	B67A	DA	99	B6 JC LBF050 ;NO, TOO SMALL, EXIT	
10606	B67D	FE	3A	. CPI 72Q ;>9 ??	
10607	B67F	D2	99	B6 JNC LBF050 ;YES, TOO BIG, EXIT	
10608	B682	.	.	. ; HAVE 0-9, GET NEXT CHAR	
10609	B682	.	.	. LBF025 EQU \$	
10610	B682	2B	.	. DCX H ;MOVE BACKWARDS IN DSP MEM	
10611	B683	7E	.	. MOV A,M ;GET NEXT CHAR	
10612	B684	C3	73	B6 JMP LBF010 ;EXAMINE IT	
10613	B687	.	.	. LBF030 EQU \$	
10614	B687	.	.	. ; HAVE DECIMAL POINT, SET FLAG	
10615	B687	0E	20	. MVI C,DECPNT	
10616	B689	.	.	. ; NOW COUNT THE NO. OF DIGITS AFTER THE DECIMAL	
10617	B689	.	.	. LBF040 EQU \$	
10618	B689	2B	.	. DCX H ;GET NEXT CHAR	
10619	B68A	7E	.	. MOV A,M	
10620	B68B	FE	30	. CPI 60Q ;< 0 ??	
10621	B68D	DA	99	B6 JC LBF050 ;YES,EXIT	
10622	B690	FE	3A	. CPI 72Q ;> 9 ??	
10623	B692	D2	99	B6 JNC LBF050 ;YES, EXIT	
10624	B695	04	.	. INR B ;UPDATE DIGIT COUNT	
10625	B696	C3	89	B6 JMP LBF040 ;GET NEXT	
10626	B699	.	.	. LBF050 EQU \$	
10627	B699	.	.	. ; IF FIELD IS EXPONENTIAL FORMAT WITH NEGATIVE	
10628	B699	.	.	. ; VALUE, DO NOT ATTEMPT TO ROUND OR REFORMAT	
10629	B699	.	.	. ; LATER ON	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 286
=====
10630    B699    FE 45 .      CPI 105Q      ;IS CHAR AN E?
10631    B69B    C2 A9 B6     JNZ LBF060    ;NO
10632    B69E    2B . .      DCX H         ;YES, GET NEXT
10633    B69F    7E . .      MOV A,M
10634    B6A0    FE 2D .      CPI 55Q      ;IS IT - ?
10635    B6A2    C2 A9 B6     JNZ LBF060    ;NO
10636    B6A5    . . .      ; FIELD IS -EXP, FORCE USE OF DEFAULT FORMAT
10637    B6A5    06 08 .      MVI B,8
10638    B6A7    0E 20 .      MVI C,DECPNT
10639    B6A9    . . .      LBF060 EQU $
10640    B6A9    . . .      ; B = NUMBER OF PLACES AFTER DECIMAL TO USE
10641    B6A9    21 EF FA     LXI H,CHRCNT ;STORE DECIMAL COUNT
10642    B6AC    70 . .      MOV M,B
10643    B6AD    3E 20 .      MVI A,DECPNT ;CLEAR PREVIOUS DEC POINT
10644    B6AF    CD C2 B8     CALL CLAPP2   ;FLAG
10645    B6B2    B1 . .      URA C         ;MERGE IN NEW
10646    B6B3    77 . .      MOV M,A
10647    B6B4    C9 . .      RET
=====
  
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 287
10649	B6B5	. . .	;*****	
10650	B6B5	. . .	; LBLRND--ROUND LABEL VALUE OFF TO NO. OF	
10651	B6B5	. . .	; DIGITS SPECIFIED IN CHRCNT. ALSO, IF	
10652	B6B5	. . .	; EXP, AND COULD WRITE IN REGULAR FORMAT	
10653	B6B5	. . .	; (.01 TO .0000001) CHANGE FORMAT	
10654	B6B5	. . .	;	
10655	B6B5	. . .	; ENTRY NUMBUF+40 = FIRST CHAR OF LABEL	
10656	B6B5	. . .	;*****	
10657	B6B5	. . .	LBLRND EQU \$	
10658	B6B5	. . .	; SET INITIAL LABEL LOCATION	
10659	B6B5	21 35 FB	LXI H,NUMBUF+40 ;WHERE OU LEFT LABEL	
10660	B6B8	22 F0 FA	SHLD LBLPTR	
10661	B6B8	. . .	; IF USER WANTS 7 OR MORE PLACES AFTER DECIMAL,	
10662	B6B8	. . .	; DONT TRY TO ROUND OR REFORMAT, JUST USE LABEL AS	
10663	B6B8	. . .	; IS	
10664	B6B8	3A EF FA	LDA CHRCNT ;7 OR MORE PLACES?	
10665	B6BE	FE 07 .	CPI 7	
10666	B6C0	D0 . .	RNC ;YES, DONE	
10667	B6C1	. . .	; IF IN EXP. FORMAT, AND IN RANGE E-2 TO E-7,	
10668	B6C1	. . .	; TRY TO CONVERT TO DECIMAL FORMAT	
10669	B6C1	CD 44 B7	CALL EXPCVT ;DO POSSIBLE CONVERSION	
10670	B6C4	2A F0 FA	LHLD LBLPTR ;POINTER TO FIRST CHAR	
10671	B6C7	. . .	; LOOK FOR DECIMAL POINT	
10672	B6C7	. . .	LBR010 EQU \$	
10673	B6C7	7E . .	MOV A,M ;FETCH CHAR	
10674	B6C8	23 . .	INX H	
10675	B6C9	FE FE .	CPI 376Q ;IS IT A DEC POINT?	
10676	B6CB	C2 C7 B6	JNZ LBR010 ;NO, KEEP LOOKING	
10677	B6CE	. . .	; HAVE DEC POINT, HL = POINTER TO FIRST CHAR AFTER	
10678	B6CE	. . .	; MOVE POINTER PAST DEC POINT BY NO. OF SIGNIFICAN	
10679	B6CE	. . .	; DIGITS IN CHRCNT (+1). IF GO PAST END OF STRING	
10680	B6CE	. . .	; DONT TRY TO ROUND	
10681	B6CE	3A EF FA	LDA CHRCNT ;FETCH DIGITS AFTER DEC COUN	
10682	B6D1	3C . .	INR A ;COUNT IN B	
10683	B6D2	47 . .	MOV B,A	
10684	B6D3	. . .	LBR020 EQU \$	
10685	B6D3	7E . .	MOV A,M ;FETCH CHAR	
10686	B6D4	FE 0A .	CPI 10 ;IS IT A DIGIT (0-9) ?	
10687	B6D6	D0 . .	RNC ;NO,CANT ROUND, PAST END	
10688	B6D7	05 . .	DCR B ;POINTER AT PROPER PLACE YET	
10689	B6D8	CA DF B6	JZ LBR030 ;YES, DO THE ADDITION	
10690	B6DB	23 . .	INX H ;NO, GET NEXT CHAR	
10691	B6DC	C3 D3 B6	JMP LBR020	
10692	B6DF	. . .	; TO ROUND, ADD 5 TO VALUE POINTED AT BY	
10693	B6DF	. . .	; HL. THIS IS ONE DIGIT LESS THAN THE LSB AS	
10694	B6DF	. . .	; SPECIFIED BY THE CHRCNT. AFTER ADDING 5,	
10695	B6DF	. . .	; PROPAGATE CY THRU REST OF STRING	
10696	B6DF	. . .	LBR030 EQU \$	
10697	B6DF	0E 05 .	MVI C,5 ;FIRST VALUE TO ADD IS 5	
10698	B6E1	. . .	LBR040 EQU \$	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 288
10699	B6E1	7E . .	MOV A,M ;FETCH CHAR	
10700	B6E2	FE FE .	CPI 376Q ;IS IT A DECIMAL POINT?	
10701	B6E4	CA FD B6	JZ LBR050 ;YES, GET NEXT CHAR	
10702	B6E7	. . .	; STRING IS STARTED BY SPACE OR - SIGN	
10703	B6E7	FE 0A .	CPI 10 ;IS IT A DIGIT?	
10704	B6E9	D2 01 B7	JNC LBR060 ;NO, STRING IS DONE	
10705	B6EC	B7 . .	ORA A ;CLEAR CARRY	
10706	B6ED	81 . .	ADD C ;ADD MEMORY VALUE	
10707	B6EE	27 . .	DAA ;CONVERT TO BCD	
10708	B6EF	4F . .	MOV C,A ;SAVE IN C	
10709	B6F0	E6 0F .	ANI 17Q ;SEND 4 LSB BACK TO SAME SPO	
10710	B6F2	77 . .	MOV M,A ;THAT DIGIT CAME FROM	
10711	B6F3	3E F0 .	MVI A,360Q ;SEE IF THERE WAS A CARRY	
10712	B6F5	A1 . .	ANA C ;OUT	
10713	B6F6	0E 00 .	MVI C,0 ;ASSUME NO	
10714	B6F8	CA FD B6	JZ LBR050 ;LEAVE C = 0 IF NO CARRY	
10715	B6FB	0E 01 .	MVI C,1 ;SET C TO CARRY	
10716	B6FD	. . .	LBR050 EQU \$	
10717	B6FD	2B . .	DCX H ;POINTER TO NEXT CHAR	
10718	B6FE	C3 E1 B6	JMP LBR040	
10719	B701	. . .	LBR060 EQU \$	
10720	B701	. . .	; HAVE ADDED PROPER VALUE TO STRING	
10721	B701	. . .	; COULD HAVE CARRY OUT OF MOST SIGNIFICANT	
10722	B701	. . .	; DIGIT. HL IS POINTING TO FIRST CHAR, EITHER	
10723	B701	. . .	; SPACE OR - SIGN	
10724	B701	47 . .	MOV B,A ;SAVE SPACE OR - IN B	
10725	B702	AF . .	XRA A ;WAS THERE AN OVERFLOW?	
10726	B703	B1 . .	ORA C ;THERE WASNT IF C = 0	
10727	B704	CA 0A B7	JZ LBR070 ;NO THERE WASNT	
10728	B707	. . .	; OVERFLOW, STORE NEW DIGIT WHERE - OR SPACE WAS	
10729	B707	71 . .	MOV M,C ;STORE OVERFLOW	
10730	B708	2B . .	DCX H	
10731	B709	70 . .	MOV M,B ;STORE SPACE OR -	
10732	B70A	. . .	LBR070 EQU \$	
10733	B70A	22 F0 FA	SHLD LBLPTR ;STORE POINTER TO START OF L	
10734	B70D	. . .	; IF NOT AN EXPONENTIAL FIELD, DONE	
10735	B70D	21 3E FB	LXI H,NUMBUF+40+9 ;WHERE E WOULD BE	
10736	B710	7E . .	MOV A,M ;IS THERE AN E THERE?	
10737	B711	FE 15 .	CPI 25Q ;CAP E?	
10738	B713	C0 . .	RNZ ;NO, DONE	
10739	B714	. . .	; IF THERE WAS AN OVERFLOW IN AN EXPONENTIAL FLD	
10740	B714	. . .	; MUST MOVE DECIMAL POINT LEFT ONE PLACE AND	
10741	B714	. . .	; UPDATE EXPONENT. IF EXP IS -, SUBTRACT ONE, AND	
10742	B714	. . .	; IF +, ADD ONE	
10743	B714	79 . .	MOV A,C ;WAS THERE AN OVERFLOW	
10744	B715	B7 . .	ORA A	
10745	B716	C8 . .	RZ ;NO, LEAVE STRING AS IS	
10746	B717	. . .	; MOVE DECIMAL POINT LEFT ONE	
10747	B717	21 36 FB	LXI H,NUMBUF+40+1 ;WHERE DEC POINT WILL GO	
10748	B71A	. . .	; HL POINTS TO SINGLE DIGIT AFTER SIGN	

=====					PAGE 289	
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	
=====						
10749	B71A	7E	.	.	MOV A,M	;FETCH DIGIT THERE
10750	B71B	36	FE	.	MVI M,376Q	;STORE A DECIMAL POINT
10751	B71D	23	.	.	INX H	;POINTER TO WHERE DEC PNT WA
10752	B71E	77	.	.	MOV M,A	;STORE CHAR THERE
10753	B71F	.	.	.	; MUST ADD 1 TO EXP IF IT IS +, SUB 1 IF -	
10754	B71F	21	3F	FB	LXI H,NUMBUF+40+10	;WHERE EXP SIGN IS
10755	B722	7E	.	.	MOV A,M	;CHECK SIGN
10756	B723	FE	FD	.	CPI 375Q	;IS IT -?
10757	B725	23	.	.	INX H	
10758	B726	23	.	.	INX H	;(PTR TO LS DIGIT OF EXP)
10759	B727	CA	38	B7	JZ LBR080	;YES, SUBTRACT 1
10760	B72A	.	.	.	; + EXPONENT, MUST ADD ONE	
10761	B72A	.	.	.	; DONT HAVE TO REFORMAT + EXPONENTIAL FORMAT	
10762	B72A	7E	.	.	MOV A,M	;FETCH DIGIT
10763	B72B	3C	.	.	INR A	;ADD ONE
10764	B72C	27	.	.	DAA	;ADJUST
10765	B72D	4F	.	.	MOV C,A	;SAVE RESULT
10766	B72E	E6	0F	.	ANI 17Q	;SEND LOWER CHAR TO MEM
10767	B730	77	.	.	MOV M,A	;NOW SEE IF CARRY TO MSDIGIT
10768	B731	3E	F0	.	MVI A,360Q	;CARRY?
10769	B733	A1	.	.	ANA C	
10770	B734	C8	.	.	RZ	;NO, DONE
10771	B735	2B	.	.	DCX H	;POINTER TO MSDIGIT
10772	B736	34	.	.	INR M	;ADD CARRY
10773	B737	C9	.	.	RET	
10774	B738	.	.	.	; SUBTRACT ONE FROM NEGATIVE EXPONENT	
10775	B738	.	.	.	LBR080 EQU \$	
10776	B738	7E	.	.	MOV A,M	;FETCH LS DIGIT
10777	B739	3D	.	.	DCR A	;SUBTRACT--WAS A 0 BEFORE?
10778	B73A	FA	3F	B7	JM LBR090	;YES, MUST BORROW
10779	B73D	77	.	.	MOV M,A	;NO, STOP UPDATED LSDIGIT
10780	B73E	C9	.	.	RET	;DONE
10781	B73F	.	.	.	LBR090 EQU \$	
10782	B73F	.	.	.	; SUBTRACTED PAST 0, SET LSDIGIT TO 9, DECREMENT	
10783	B73F	.	.	.	; MSDIGIT	
10784	B73F	36	09	.	MVI M,9	;STORE NEW LSDIGIT
10785	B741	2B	.	.	DCX H	;POINTER TO MSDIGIT
10786	B742	35	.	.	DCR M	
10787	B743	C9	.	.	RET	;DONE

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 290
10789	B744	.	.	*****	
10790	B744	.	.	; EXPCVT--CONVERT FROM EXPONENTIAL TO DECIMAL	
10791	B744	.	.	; NOTATION IF EXP IS IN RANGE E-02 TO E-07 AND	
10792	B744	.	.	; NO. OF LEADING ZEROS + NUMBER OF SIGNIFICANT	
10793	B744	.	.	; DIGITS IS .LE.7	
10794	B744	.	.	;	
10795	B744	.	.	; ENTRY NUMBUF + 40 = START OF STRING	
10796	B744	.	.	*****	
10797	B744	.	.	EXPCVT EQU \$	
10798	B744	21	3E FB	LXI H,NUMBUF+40+9 ;WHERE E WOULD BE	
10799	B747	7E	.	MOV A,M ;IS THERE AN E THERE	
10800	B748	FE	15 .	CPI 250 ;CAP E?	
10801	B74A	C0	.	RNZ ;NO, CANT CONVERT	
10802	B74B	23	.	INX H	
10803	B74C	7E	.	MOV A,M ;IS FIELD - ?	
10804	B74D	FE	FD .	CPI 3750	
10805	B74F	C0	.	RNZ ;NO, CANT CONVERT	
10806	B750	23	.	INX H	
10807	B751	7E	.	MOV A,M ;MSDIGIT 0?	
10808	B752	B7	.	ORA A	
10809	B753	C0	.	RNZ ;NO, CANT CONVERT	
10810	B754	23	.	INX H	
10811	B755	7E	.	MOV A,M ;IS LSDIGIT .LE. 7?	
10812	B756	FE	08 .	CPI 8	
10813	B758	D0	.	RNC ;NO, CANT CONVERT	
10814	B759	3D	.	DCR A ;NO OF LEADING ZEROS	
10815	B75A	C8	.	RZ ;CANT CONVERT IF LSDIGIT	
10816	B75B	F8	.	RM ;WAS 1 OR 0	
10817	B75C	4F	.	MOV C,A ;C = LEADING 0 COUNT	
10818	B75D	.	.	; XFER STRING TO TEMP BUFFER -- ADD THE ZEROS	
10819	B75D	21	35 FB	LXI H,NUMBUF+40 ;START OF STRING	
10820	B760	11	49 FB	LXI D,NUMBUF+60 ;DESTINATION	
10821	B763	7E	.	MOV A,M ;FETCH SIGN	
10822	B764	12	.	STAX D	
10823	B765	13	.	INX D	
10824	B766	3E	FE .	MVI A,3760 ;STORE A DECIMAL POINT	
10825	B768	12	.	STAX D	
10826	B769	.	.	; STORE LEADING ZEROS	
10827	B769	AF	.	XRA A ;SET A FOR 0	
10828	B76A	.	.	EXC010 EQU \$	
10829	B76A	13	.	INX D ;UPDATE DESTINATION	
10830	B76B	12	.	STAX D ;STORE A ZERO	
10831	B76C	0D	.	DCR C ;ALL ZEROS STORED	
10832	B76D	C2	6A B7	JNZ EXC010 ;NO, CONTINUE LOOPING	
10833	B770	.	.	; NOW TRANSFER ALL THE REMAINING DIGITS	
10834	B770	06	07 .	MVI B,7 ;7 DIGITS LEFT	
10835	B772	.	.	EXC030 EQU \$	
10836	B772	23	.	INX H ;POINTER TO NEXT DIGIT	
10837	B773	7E	.	MOV A,M ;FETCH DIGIT	
10838	B774	FE	FE .	CPI 3760 ;IS IT A DEC POINT?	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 291
=====
10839    B776     CA  72  B7          JZ   EXC030      ;YES, IGNORE
10840    B779     13  .   .          INX  D          ;DESTINATION POINTER
10841    B77A     12  .   .          STAX D         ;STORE DIGIT
10842    B77B     05  .   .          DCR  B         ;ALL DIGITS STORED?
10843    B77C     C2  72  B7          JNZ  EXC030
10844    B77F     .   .   .          ; NOW TRANSFER NEW STRING BACK TO OLD LOCATION
10845    B77F     21  35  FB          LXI  H,NUMBUF+40 ;DESTINATION
10846    B782     11  49  FB          LXI  D,NUMBUF+60 ;SOURCE
10847    B785     0E  0A  .          MVI  C,10      ;ONLY DO FIRST 10 CHARS
10848    B787     .   .   .          EXC040 EQU $
10849    B787     1A  .   .          LDAX D         ;FETCH CHAR
10850    B788     77  .   .          MOV  M,A'      ;STORE IT
10851    B789     23  .   .          INX  H
10852    B78A     13  .   .          INX  D
10853    B78B     0D  .   .          DCR  C         ;ALL 10 DONE?
10854    B78C     C2  87  B7          JNZ  EXC040    ;NO
10855    B78F     .   .   .          ; PUT 3 SPACES AT END OF STRING
10856    B78F     0E  03  .          MVI  C,3
10857    B791     .   .   .          EXC050 EQU $
10858    B791     36  F0  .          MVI  M,3600   ;STORE A SPACE
10859    B793     23  .   .          INX  H
10860    B794     0D  .   .          DCR  C         ;ALL DONE?
10861    B795     C2  91  B7          JNZ  EXC050    ;NO, CONTINUE LOOPING
10862    B798     C9  .   .          RET
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 292
10864	B799	. . .	;*****	
10865	B799	. . .	; APLTON--TURN AUTO PLOT ON	
10866	B799	. . .	;*****	
10867	B799	. . .	APLTON EQU \$	
10868	B799	CD C6 00	CALL ZCHKSF ;SOFT KEYS UP?	
10869	B79C	C0 . .	RNZ ;YES, DONT DO AUTO PLOT	
10870	B79D	. . .	; IF ANY OF NUMBER OF COLS, X COL, OR Y COL IS 0,	
10871	B79D	. . .	; DONT TURN AUTO PLOT ON	
10872	B79D	3A C3 FB	LDA NMCLBF ;NO. OF COLS = 0?	
10873	B7A0	B7 . .	ORA A	
10874	B7A1	C8 . .	RZ	
10875	B7A2	3A C2 FB	LDA XCOLBF ;X COL = 0?	
10876	B7A5	B7 . .	ORA A	
10877	B7A6	C8 . .	RZ	
10878	B7A7	3A C1 FB	LDA YCOLBF ;Y COL = 0?	
10879	B7AA	B7 . .	ORA A	
10880	B7AB	C8 . .	RZ	
10881	B7AC	CD 69 B9	CALL RNGCHK ;MIN, MAX VALUES OK?	
10882	B7AF	F2 8D B9	JP APERR ;NO, REPORT ERROR	
10883	B7B2	CD 1E 76	CALL GTXOF1 ;G TEXT OFF	
10884	B7B5	CD 02 70	CALL TGCOF1 ;CURSOR OFF	
10885	B7B8	CD D9 AA	CALL APMUOF ;MENU OFF	
10886	B7BB	CD AC 6D	CALL GVON1 ;TURN THE GRAPHICS VIDEO ON	
10887	B7BE	AF . .	XRA A ;CLEAR COLUMN COUNT	
10888	B7BF	32 0C FB	STA COLCNT	
10889	B7C2	32 96 FB	STA APFLGS ;CLEAR AUTO PLOT FLAGS	
10890	B7C5	32 97 FB	STA APFLG2	
10891	B7C8	32 05 FB	STA IGNCNT ;RESET IGNORE COUNT	
10892	B7CB	3E 02 .	MVI A,APIP ;SET AUTO PLOT IN PROGRESS	
10893	B7CD	CD AF B8	CALL STAPFL	
10894	B7D0	. . .	; COMPUTE SCALE FACTORS	
10895	B7D0	CD 73 AA	CALL APINIT	
10896	B7D3	. . .	; PUT INTO SET MODE	
10897	B7D3	3E 02 .	MVI A,2	
10898	B7D5	CD 20 72	CALL SETMD1	
10899	B7D8	. . .	; SET AUTO PLOT CLIPPING LIMITS	
10900	B7D8	21 BA FF	LXI H,-XOFSET	
10901	B7DB	22 72 90	SHLD XMIN	
10902	B7DE	21 D3 FF	LXI H,-YOFSET	
10903	B7E1	22 6E 90	SHLD YMIN	
10904	B7E4	21 54 FD	LXI H,-XOFSET-XAXLEN	
10905	B7E7	22 70 90	SHLD XMAX	
10906	B7EA	21 A3 FE	LXI H,-YOFSET-YAXLEN	
10907	B7ED	22 6C 90	SHLD YMAX	
10908	B7F0	CD 40 63	CALL HRD2 ;COMPUTE NEW BOUNDS CODE	
10909	B7F3	CD 4F 00	CALL ZESCND ;TERMINATE ANY PENDING ESC S	
10910	B7F6	3E 01 .	MVI A,MOVE ;FIRST VECTOR OF PLOT	
10911	B7F8	. . .	;*****	
10912	B7F8	. . .	; ROM BREAK 8	
10913	B7FB	C3 02 B8	JMP ZBRK8C	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS
10914	B7FB	.	.	.	ORG ZBRK7+4000Q
10915	B800	.	.	.	ZBRK8 EQU \$
10916	B800	54	.	.	DB VERSN
10917	B801	B8	.	.	DB ZBRK8/256
10918	B802	.	.	.	ZBRK8C EQU \$
10919	B802	.	.	.	;*****
10920	B802	CD	26	A2	CALL STFLG1 ;IS A MOVE
10921	B805	.	.	.	; LOAD POINT COUNT, SKIP COUNT AND LINE TYPE
10922	B805	.	.	.	; FROM MENU
10923	B805	2A	9C	FB	LHLD CNTBF ;FETCH NO. OF POINTS TO PLOT
10924	B808	2B	.	.	DCX H
10925	B809	22	08	FB	SHLD PNTCNT
10926	B80C	2A	9E	FB	LHLD SKPBF ;FETCH NO. OF LINES TO SKIP
10927	B80F	2B	.	.	DCX H
10928	B810	22	0A	FB	SHLD SKPCNT
10929	B813	.	.	.	; LOAD THE LINE TYPE
10930	B813	3A	C0	FB	LDA LINEBF ;FETCH THE SELECTION
10931	B816	3D	.	.	DCR A ;WANT 0-4
10932	B817	.	.	.	; IF LINE TYPE 0 IS SELECTED, USE THE
10933	B817	.	.	.	; CURRENT LINE TYPE
10934	B817	F4	64	72	CP SETLN2 ;TURN IT ON
10935	B81A	.	.	.	; NOW SEE IF DISPLAY MEMORY IS TO BE PLOTTED
10936	B81A	2A	98	FB	LHLD FROMBF ;NON ZERO ENTRY IN
10937	B81D	7C	.	.	MOV A,H ;'FROM DISPLAY?' MENU
10938	B81E	B5	.	.	ORA L ;FIELD?
10939	B81F	C8	.	.	RZ ;NO, DONT PLOT FROM DISLAY
10940	B820	.	.	.	; PLOT DATA IN DISPLAY MEMORY
10941	B820	3E	02	.	MVI A,APDISP ;SET PLOTTING FROM DISPLAY
10942	B822	CD	BC	B8	CALL STAFP2 ;FLAG
10943	B825	.	.	.	; PLOT THE DATA IN DISPLAY MEMORY
10944	B825	.	.	.	; TURN AUTO PLOT OFF AFTER FINISHED
10945	B825	CD	E4	00	CALL ZINITG ;INITIALIZE FOR GET DISPLAY
10946	B828	C2	82	B8	JNZ APLTOF ;STOP IF NO CHARACTERS
10947	B82B	21	64	FF	LXI H,ZIOFL2 ;DISABLE EXPANSION OF ESC
10948	B82E	3E	20	.	MVI A,ZDS2BF ;SEGS BY GETDSP
10949	B830	B6	.	.	ORA M
10950	B831	77	.	.	MOV M,A
10951	B832	.	.	.	APL005 EQU \$
10952	B832	.	.	.	;INSERT IS UPDATED BY THE 2645 MAIN CODE ROUTINE
10953	B832	.	.	.	; DISPC0 WHEN DISPLAY CODES ARE ADDED BY HILITE
10954	B832	AF	.	.	XRA A ;RESET INSERTED CHARACTER
10955	B833	32	04	FB	STA INSERT ;COUNT
10956	B836	.	.	.	APL010 EQU \$
10957	B836	21	96	FB	LXI H,APFLGS ;STILL PLOTTING?
10958	B839	46	.	.	MOV B,M
10959	B83A	3E	02	.	MVI A,APIP
10960	B83C	A0	.	.	ANA B
10961	B83D	CA	7B	B8	JZ APL060 ;NO, TERNMINATE AUTO PLOT
10962	B840	.	.	.	APL020 EQU \$
10963	B840	C5	.	.	PUSH B ;SAVE FLAGS IN B

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 294
=====
10964    B841      .      .      .      ; GETDSP UPDATES CURSOR POSITION, WHICH SCREWS
10965    B841      .      .      .      ; UP HILITE SUBR AND END OF LINE PROCESSING
10966    B841      2A    C0    FF      LHL D ZCUROW      ;SO, SAVE THE CURRENT
10967    B844      22    06    FB      SHLD CURSAV      ;CURSOR POSITION
10968    B847      CD    88    00      CALL ZGETDP      ;GET NEXT CHAR FROM DISPLAY
10969    B84A      C1      .      .      POP B            ;RECALL FLAGS IN B
10970    B84B      2A    C0    FF      LHL D ZCUROW      ;SAVE UPDATED CURSOR
10971    B84E      E5      .      .      PUSH H          ;POSITION
10972    B84F      2A    06    FB      LHL D CURSAV     ;RECALL ORIGINAL POSITION
10973    B852      22    C0    FF      SHLD ZCUROW     ;BEFORE GETDSP CHANGED IT
10974    B855      D2    6B    B8      JNC APL030      ;CHAR FOUND, PROCESS IT
10975    B858      FA    7A    B8      JM APL050       ;STOP--END OF DISPLAY
10976    B85B      .      .      .      ;END OF LINE--STOP BUILDING NUMBER, DO LINE FEED
10977    B85B      3E    01    .      MVI A,NIP      ;WAS A NUMBER BEING BUILT?
10978    B85D      A0      .      .      ANA B
10979    B85E      C4    B4    B0      CNZ STOP        ;IF YES, TERMINATE IT
10980    B861      E1      .      .      POP H            ;RESTORE UPDATED CURSOR
10981    B862      22    C0    FF      SHLD ZCUROW     ;POSITION
10982    B865      CD    8B    00      CALL ZLNFD      ;DO A LINE FEED
10983    B868      C3    32    B8      JMP APL005      ;GET NEXT CHAR
10984    B868      .      .      .      ;CHARACTER FOUND
10985    B86B      .      .      .      APL030 EQU $
10986    B86B      B7      .      .      ORA A           ;IS IT DISPLAYABLE??
10987    B86C      FA    73    B8      JM APL040      ;NO, DONT PROCESS IT
10988    B86F      4F      .      .      MOV C,A        ;LEAVE CHAR IN C
10989    B870      CD    87    AF      CALL APSCN1    ;DO AUTO PLOT SCAN
10990    B873      J      .      .      APL040 EQU $
10991    B873      E1      .      .      POP H            ;RESTORE UPDATED CURSOR
10992    B874      22    C0    FF      SHLD ZCUROW     ;POSITION
10993    B877      C3    36    B8      JMP APL010     ;GET NEXT CHAR
10994    B87A      .      .      .      ;END OF DISPLAY--STOP PLOTTING
10995    B87A      .      .      .      APL050 EQU $
10996    B87A      E1      .      .      POP H            ;RESTORE STACK
10997    B87B      .      .      .      APL060 EQU $
10998    B87B      21    64    FF      LXI H,ZIOFL2   ;CLEAR EXPANSION INHIBIT
10999    B87E      3E    DF    .      MVI A,-1-ZDS2BF ;FLAG
11000    B880      A6      .      .      ANA M
11001    B881      77      .      .      MOV M,A        ;FALL INTO AUTO PLOT OFF
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 295
11003	B882	.	.	*****	
11004	B882	.	.	; APLTOF--TURN AUTO PLOT OFF	
11005	B882	.	.	*****	
11006	B882	.	.	APLTOF EQU \$	
11007	B882	21	96	FB LXI H,APFLGS ;IS AUTO PLOT NOW?	
11008	B885	3E	02	. MVI A,APIP	
11009	B887	A6	.	. ANA M	
11010	B888	C8	.	. RZ ;NO,EXIT	
11011	B889	36	00	. MVI M,0 ;YES, CLEAR ALL FLAGS	
11012	B88B	.	.	. ; RESTORE NORMAL CLIP LIMITS	
11013	B88B	CD	2B	63 CALL HRD1	
11014	B88E	3E	FE	. MVI A,-1-MENUON ;CLEAR OTHR AUTO PLOT FLGS	
11015	B890	CD	C2	B8 CALL CLAPF2	
11016	B893	.	.	. ; TURN BLINKING GOLD LED OFF IF NOT	
11017	B893	.	.	. ; IN SELECT MODE	
11018	B893	3A	F4	FF LDA ZMDFL1 ;IN SELECT MODE?	
11019	B896	E6	20	. ANI SELECT	
11020	B898	C0	.	. RNZ ;YES, LEAVE GOLD LED ON	
11021	B899	.	.	. ; TURN THE GOLD LED OFF	
11022	B899	3E	DF	. MVI A,-1-SELLED	
11023	B89B	4F	.	. MOV C,A ;A=C=MASK TO CLEAR LED	
11024	B89C	.	.	. ; CLEAR THE BLINK FLAG	
11025	B89C	21	0E	FF LXI H,ZBLFLG	
11026	B89F	A6	.	. ANA M	
11027	B8A0	77	.	. MOV M,A	
11028	B8A1	.	.	. ; TURN THE GOLD LED OFF	
11029	B8A1	21	0C	FF LXI H,ZKBLED	
11030	B8A4	F3	.	. DI ;DONT WANT TIMER INTERRUPTS	
11031	B8A5	7E	.	. MOV A,M	
11032	B8A6	FE	7F	. CPI 1770 ;ALL LEDS ON => SELF TEST	
11033	B8A8	CA	AD	B8 JZ APO010 ;IF SO, LEAVE IT ON	
11034	B8AB	A1	.	. ANA C	
11035	B8AC	77	.	. MOV M,A	
11036	B8AD	.	.	. APO010 EQU \$	
11037	B8AD	FB	.	. EI	
11038	B8AE	C9	.	. RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 296
11040	B8AF	.	.	*****	
11041	B8AF	.	.	; STAPFL--SET AUTO PLOT FLAG	
11042	B8AF	.	.	; ENTRY A = FLAG(S)	
11043	B8AF	.	.	*****	
11044	B8AF	.	.	STAPFL EQU \$	
11045	B8AF	21	96	FB LXI H,APFLGS	
11046	B8B2	B6	.	ORA M	
11047	B8B3	77	.	MOV M,A	
11048	B8B4	C9	.	RET	
11049	B8B5	.	.	*****	
11050	B8B5	.	.	; CLAPFL--CLEAR AUTO PLOT FLAG	
11051	B8B5	.	.	; ENTRY A = FLAG(S)	
11052	B8B5	.	.	*****	
11053	B8B5	.	.	CLAPFL EQU \$	
11054	B8B5	21	96	FB LXI H,APFLGS	
11055	B8B8	2F	.	CMA	
11056	B8B9	A6	.	ANA M	
11057	B8BA	77	.	MOV M,A	
11058	B8BB	C9	.	RET	
11059	B8BC	.	.	*****	
11060	B8BC	.	.	; STAPF2--SET FLAG IN APFLG2	
11061	B8BC	.	.	*****	
11062	B8BC	.	.	STAPF2 EQU \$	
11063	B8BC	21	97	FB LXI H,APFLG2	
11064	B8BF	B6	.	ORA M	
11065	B8C0	77	.	MOV M,A	
11066	B8C1	C9	.	RET	
11067	B8C2	.	.	*****	
11068	B8C2	.	.	; CLAPF2--CLEAR FLAG IN APFLG2	
11069	B8C2	.	.	*****	
11070	B8C2	.	.	CLAPF2 EQU \$	
11071	B8C2	21	97	FB LXI H,APFLG2	
11072	B8C5	2F	.	CMA	
11073	B8C6	A6	.	ANA M	
11074	B8C7	77	.	MOV M,A	
11075	B8C8	C9	.	RET	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS
11077	B8C9	. . .	;*****
11078	B8C9	. . .	; INT--CONVERT VALUE IN FLOATING POINT
11079	B8C9	. . .	; ACCUMULATOR TO ROUNDED INTEGER
11080	B8C9	. . .	; EXIT--HL = ROUNDED INTEGER (+ OR -)
11081	B8C9	. . .	;*****
11082	B8C9	. . .	INT EQU \$
11083	B8C9	1E 1F .	MVI E,31 ;CONVERT TO 2 * VALUE
11084	B8CB	CD 17 BB	CALL FIX
11085	B8CE	. . .	; FIX LEAVES TWOS COMPLEMENT RESULT IN ABCD
11086	B8CE	. . .	; REGISTERS, A=MSBYTE, D=LSBYTE
11087	B8CE	. . .	; TO ROUND, ADD 1, AND SHIFT RIGHT ONCE MORE
11088	B8CE	. . .	; THIS IS EQUIVALENT TO ADDING .5
11089	B8CE	. . .	; IF ABCD IS +, ADD +1. IF -, ADD -1
11090	B8CE	21 01 00	LXI H,1 ;ASSUME +
11091	B8D1	B7 . .	ORA A ;TEST SIGN
11092	B8D2	F2 D8 B8	JP INT010 ;REALLY IS +
11093	B8D5	21 FF FF	LXI H,-1 ;IS -
11094	B8D8	. . .	INT010 EQU \$
11095	B8D8	5F . .	MOV E,A ;SAVE A REG IN E
11096	B8D9	7D . .	MOV A,L ;ADD 1 TO LSBYTE
11097	B8DA	82 . .	ADD D
11098	B8DB	57 . .	MOV D,A
11099	B8DC	. . .	; PROPAGATE CARRY THROUGH C,B,A
11100	B8DC	7C . .	MOV A,H
11101	B8DD	89 . .	ADC C
11102	B8DE	4F . .	MOV C,A
11103	B8DF	7C . .	MOV A,H
11104	B8E0	88 . .	ADC B
11105	B8E1	47 . .	MOV B,A
11106	B8E2	7C . .	MOV A,H
11107	B8E3	8B . .	ADC E ;MSBYTE DONE
11108	B8E4	. . .	; NOW SHIFT A B C D RIGHT ONE
11109	B8E4	1F . .	RAR
11110	B8E5	78 . .	MOV A,B
11111	B8E6	1F . .	RAR
11112	B8E7	79 . .	MOV A,C
11113	B8E8	1F . .	RAR
11114	B8E9	67 . .	MOV H,A ;LEAVE RESULT IN HL
11115	B8EA	7A . .	MOV A,D
11116	B8EB	1F . .	RAR
11117	B8EC	6F . .	MOV L,A
11118	B8ED	C9 . .	RET

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 298
11120	B8EE	.	.	*****	
11121	B8EE	.	.	; MUCHK-- SEE IF AUTO PLOT MENU IS ON	
11122	B8EE	.	.	; ENTRY--DONT CARE	
11123	B8EE	.	.	; EXIT NZ => AUTO PLOT MENU ON	
11124	B8EE	.	.	; A DESTROYED	
11125	B8EE	.	.	*****	
11126	B8EE	.	.	MUCHK EQU \$	
11127	B8EE	3A	97 FB	LDA APFLG2	
11128	B8F1	E6	01 .	ANI MENUON	
11129	B8F3	C9	.	RET	
11130	B8F4	.	.	*****	
11131	B8F4	.	.	; CHEKAP--SEE IF AUTO PLOT IS ON	
11132	B8F4	.	.	; ENTRY--DONT CARE	
11133	B8F4	.	.	; EXIT NZ => AUTO PLOT IS ON	
11134	B8F4	.	.	; A DESTROYED	
11135	B8F4	.	.	*****	
11136	B8F4	.	.	CHEKAP EQU \$	
11137	B8F4	3A	96 FB	LDA APFLGS	
11138	B8F7	E6	02 .	ANI APIP	
11139	B8F9	C9	.	RET	
11140	B8FA	.	.	*****	
11141	B8FA	.	.	; APCHK--PROCESS KEYBOARD CHAR SPECIAL IF IN	
11142	B8FA	.	.	; AUTO PLOT MODE	
11143	B8FA	.	.	; 1. IF NO VECTORS HAVE BEEN PLOTTED, IGNORE	
11144	B8FA	.	.	; 2. IF FIRST POINT HAS BEEN PROCESSED, STOP	
11145	B8FA	.	.	; AUTO PLOT	
11146	B8FA	.	.	; AUTO PLOT MUST BE ON WHEN CALLED	
11147	B8FA	.	.	; ENTRY C = CHAR	
11148	B8FA	.	.	; EXIT BC SAVED	
11149	B8FA	.	.	; A,HL DESTROYED	
11150	B8FA	.	.	*****	
11151	B8FA	.	.	APCHK EQU \$	
11152	B8FA	3A	B2 90	LDA GFLGS1 ;FIRST POINT PLOTTED?	
11153	B8FD	E6	01 .	ANI MOVE	
11154	B8FF	CA	13 B9	JZ APC010 ;YES, TURN AUTO PLOT OFF	
11155	B902	.	.	; HAVENT STARTED PLOTTING YET, IGNORE CHAR	
11156	B902	.	.	; IF IT IS A DIGIT THATS NOT IN AN ESC SEQ	
11157	B902	79	.	MOV A,C ;IS CHAR A DIGIT?	
11158	B903	FE	30 .	CPI 600	
11159	B905	D8	.	RC ;NO, TOO SMALL	
11160	B906	FE	3A .	CPI 720	
11161	B908	D0	.	RNC ;NO, TOO LARGE	
11162	B909	3A	D1 FF	LDA ZESCFG ;YES, IS AN ESC SEQ GOING?	
11163	B90C	B7	.	ORA A	
11164	B90D	C0	.	RNZ ;YES, IGNORE THE DIGIT	
11165	B90E	.	.	; HAVE DIGIT FROM KEYBOARD, IGNORE IT	
11166	B90E	21	05 FB	LXI H,IGNCNT ;UPDATE NO. OF CHARS TO	
11167	B911	34	.	INR M ;BE IGNORED	
11168	B912	C9	.	RET	
11169	B913	.	.	; KEYBOARD CHARACTER RECEIVED IN AUTO PLOT MODE	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 299
=====
11170    B913    . . .      ; TURN AUTO PLOT OFF
11171    B913    . . .      APC010 EQU $
11172    B913    C5 . .     PUSH B ;SAVE C REG
11173    B914    CD 82 88   CALL APLTOF ;TURN AUTO PLOT OFF
11174    B917    C1 . .     POP B
11175    B918    C9 . .     RET
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 300
11177	B919	.	.	*****	
11178	B919	.	.	; CHKCH--CHECK SOURCE OF CHARACTER. IF NOT FROM	
11179	B919	.	.	; KEYBOARD, TURN MENU OFF, ADD CHAR TO NORMAL	
11180	B919	.	.	; DISPLAY	
11181	B919	.	.	*****	
11182	B919	.	.	CHKCH EQU \$	
11183	B919	CD	C3 00	CALL ZDCIO ;FROM DC OR IO?	
11184	B91C	C8	.	RZ ;NO	
11185	B91D	.	.	CHKCH1 EQU \$	
11186	B91D	CD	D9 AA	CALL APMUOF ;YES, TURN THE MENU OFF	
11187	B920	C3	C9 00	JMP ZDSPCH ;ADD CHAR TO NORMAL DISPLAY	
11188	B923	.	.	*****	
11189	B923	.	.	; ADJCOL--IF AUTOPLOTTING FROM DISPLAY MEM,	
11190	B923	.	.	; UPDATE THE CURSOR COLUMN BY SUBTRACTING	
11191	B923	.	.	; THE NUMBER OF CHARS (DIVIDED BY 2) INSERTED	
11192	B923	.	.	; BY HILITE WHEN IT WAS HILIGHTING A NUMBER	
11193	B923	.	.	; IN INVERSE VIDEO	
11194	B923	.	.	; INSERT IS UPDATED BY DISPC0 IN 2645 MAIN CODE	
11195	B923	.	.	; ENTRY H = COL	
11196	B923	.	.	; EXIT H = NEW COL, A DESTROYED	
11197	B923	.	.	*****	
11198	B923	.	.	ADJCOL EQU \$	
11199	B923	3A	97 FB	LDA APFLG2 ;PLOTTING FROM DISPLAY?	
11200	B926	E6	02 .	ANI APDISP	
11201	B928	C8	.	RZ ;NO, LEAVE COL AS IS	
11202	B929	3A	04 FB	LDA INSERT ;FETCH INSERTED CHAR COUNT	
11203	B92C	B7	.	ORA A ;(CLEAR CARRY)	
11204	B92D	1F	.	RAR ;DIVIDE BY 2	
11205	B92E	2F	.	CMA ;CONVERT TO NEGATIVE	
11206	B92F	3C	.	INR A	
11207	B930	84	.	ADD H ;SUBTRACT FROM COLUMN	
11208	B931	67	.	MOV H,A ;NEW COLUMN INTO H	
11209	B932	C9	.	RET	
11210	B933	.	.	*****	
11211	B933	.	.	; INSFIX--CALLED BY DISPC0 IN 2645 MAIN CODE	
11212	B933	.	.	; WHEN DISPLAY ENHANCEMENT IS INSERTED BY	
11213	B933	.	.	; HILITE SUBR. USED BY ADJCOL TO COMPENSATE FOR	
11214	B933	.	.	; ADDED CHARACTERS	
11215	B933	.	.	; ENTRY--DONT CARE	
11216	B933	.	.	; EXIT---ALL REGISTERS SAVED	
11217	B933	.	.	*****	
11218	B933	.	.	INSFIX EQU \$	
11219	B933	F5	.	PUSH PSW ;SAVE FLAGS	
11220	B934	3A	04 FB	LDA INSERT	
11221	B937	3C	.	INR A ;UPDATE COUNT	
11222	B938	32	04 FB	STA INSERT	
11223	B93B	F1	.	POP PSW	
11224	B93C	C9	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 301
11226	B93D	.	.	*****	
11227	B93D	.	.	; TRUNCT--TRUNCATE FRACTIONAL PART OF FLOATING	
11228	B93D	.	.	; POINT VALUE BY FIXING, THEN REFLOATING	
11229	B93D	.	.	; DO NOT ROUND WHEN FIXING	
11230	B93D	.	.	; ADD -1 BEFORE FLOATING.	
11231	B93D	.	.	; THIS IS USED BY XTICS AND YTICS TO INSURE	
11232	B93D	.	.	; FIRST TIC IS .LE. MIN IF MIN IS -	
11233	B93D	.	.	; OR ZERO	
11234	B93D	.	.	*****	
11235	B93D	.	.	TRUNCT EQU \$	
11236	B93D	1E	20	MVI E,32	
11237	B93F	CD	17	CALL FIX ;CONVERT TO INTEGER	
11238	B942	.	.	; ADD -1 TO INTEGER	
11239	B942	1E	FF	MVI E,-1	
11240	B944	6F	.	MOV L,A ;SAVE A REG	
11241	B945	7A	.	MOV A,D ;ADD LSBYTE	
11242	B946	83	.	ADD E	
11243	B947	57	.	MOV D,A	
11244	B948	79	.	MOV A,C	
11245	B949	8B	.	ADC E ;PROPAGATE CARRY	
11246	B94A	4F	.	MOV C,A	
11247	B94B	78	.	MOV A,B	
11248	B94C	8B	.	ADC E	
11249	B94D	47	.	MOV B,A	
11250	B94E	7D	.	MOV A,L ;ADD MSBYTE	
11251	B94F	8B	.	ADC E	
11252	B950	.	.	; ADDITION COMPLETE, CONVERT BACK TO FP	
11253	B950	1E	20	MVI E,32	
11254	B952	C3	00	JMP FLT ;BACK TO FP	
11255	B955	.	.	*****	
11256	B955	.	.	; CHKMIN-- SEE IF HL IS LESS THAN SPECIFIED	
11257	B955	.	.	; VALUE IN DE	
11258	B955	.	.	; EXIT NC => HL .GE. DE (OK)	
11259	B955	.	.	; CY => HL .LT. DE (TOO SMALL)	
11260	B955	.	.	; A DESTROYED	
11261	B955	.	.	*****	
11262	B955	.	.	CHKMIN EQU \$	
11263	B955	7C	.	MOV A,H	
11264	B956	B7	.	ORA A ;SEE IF -	
11265	B957	37	.	STC ;(ASSUME YES)	
11266	B958	F8	.	RM ;YES, - TOO SMALL	
11267	B959	BA	.	CMP D ;COMPARE MSBYTES	
11268	B95A	D8	.	RC ;TOO SMALL	
11269	B95B	C0	.	RNZ ;DONT NEED TO CHECK LSBYTES	
11270	B95C	7D	.	MOV A,L ;COMPARE LSBYTES	
11271	B95D	BB	.	CMP E ;CY SET IF TOO SMALL	
11272	B95E	C9	.	RET	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 302
11274	B95F	. . .	;*****	
11275	B95F	. . .	; CHKMAX--SEE IF HL IS GREATER THAN SPECIFIED	
11276	B95F	. . .	; VALUE IN DE	
11277	B95F	. . .	; EXIT NC => HL .LE. DE (OK)	
11278	B95F	. . .	; CY => HL .GT. DE (TOO BIG)	
11279	B95F	. . .	; A DESTROYED	
11280	B95F	. . .	;*****	
11281	B95F	. . .	CHKMAX EQU \$	
11282	B95F	7C . .	MOV A,H ;IN RANGE IF NEGATIVE	
11283	B960	B7 . .	ORA A ;- ??	
11284	B961	F8 . .	RM ;YES, RETURN AS IN RANGE	
11285	B962	7A . .	MOV A,D ;COMPARE MSBYTES	
11286	B963	8C . .	CMP H	
11287	B964	D8 . .	RC ;TOO BIG	
11288	B965	C0 . .	RNZ ;NO NEED TO CHECK LSBYTES	
11289	B966	7B . .	MOV A,E ;COMPARE LSBYTES	
11290	B967	BD . .	CMP L	
11291	B968	C9 . .	RET ;CY SET IF TOO BIG	
11292	B969	. . .	;*****	
11293	B969	. . .	; RNGCHK--SEE IF XMIN .LT. XMAX,AND YMIN. LT. YMAX	
11294	B969	. . .	; EXIT ALL REGISTERS DESTROYED	
11295	B969	. . .	; P => X OR Y OUT OF RANGE	
11296	B969	. . .	; HL = POINTER TO X OR Y IF ERROR	
11297	B969	. . .	;*****	
11298	B969	. . .	RNGCHK EQU \$	
11299	B969	21 BC FB	LXI H,XMINBF ;COMPUTE XMIN-XMAX	
11300	B96C	CD 6E BD	CALL LOD	
11301	B96F	21 88 FB	LXI H,XMAXBF	
11302	B972	CD D3 BD	CALL SB	
11303	B975	21 89 B9	LXI H,APXERR ;(ASSUME X IN ERROR)	
11304	B978	F0 . .	RP ;ERROR IF RESULT +	
11305	B979	21 84 FB	LXI H,YMINBF ;COMPUTE YMIN-YMAX	
11306	B97C	CD 6E BD	CALL LOD	
11307	B97F	21 80 FB	LXI H,YMAXBF	
11308	B982	CD D3 BD	CALL SB	
11309	B985	21 88 B9	LXI H,APYERR ;(ASSUME Y IN ERROR)	
11310	B988	C9 . .	RET ;POS IF ERROR	
11311	B989	. . .	;	
11312	B989	58 00 .	APXERR DB 'X',0 ;X ERROR	
11313	B98B	59 00 .	APYERR DB 'Y',0 ;Y ERROR	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 303
=====
11315    B98D      . . .      ;*****
11316    B98D      . . .      ; APERR X OR Y VALUES BAD
11317    B98D      . . .      ; PUT ERROR MESSAGE UP, WAIT FOR RETURN HIT
11318    B98D      . . .      ; TO RESTORE DISPLAY
11319    B98D      . . .      ; ENTRY HL = POINTER TO X OR Y MESSAGE
11320    B98D      . . .      ;*****
11321    B98D      . . .      APERR EQU $
11322    B98D      22 ED FF    SHLD ZMSGP3 ;STORE X OR Y MESSAGE
11323    B990      21 B6 B9    LXI H,APMSG1 ;SET OTHER POINTERS
11324    B993      22 F1 FF    SHLD ZMSGP1
11325    B996      21 CA B9    LXI H,APMSG2
11326    B999      22 EF FF    SHLD ZMSGP2
11327    B99C      21 D3 B9    LXI H,APMSG3
11328    B99F      22 EB FF    SHLD ZMSGP4
11329    B9A2      . . .      ; PUT MESSAGE ON DISPLAY
11330    B9A2      . . .      APE010 EQU $
11331    B9A2      37 . . .    STC ;REPLACE DISPLAY WITH MESSAG
11332    B9A3      CD 40 00    CALL ZDSPMG
11333    B9A6      . . .      ; WAIT FOR CR HIT
11334    B9A6      . . .      APE1 EQU $
11335    B9A6      CD 28 A4    CALL GETKEY ;GET A KEY
11336    B9A9      FE EF .     CPI SFTCR ;RETURN KEY??
11337    B9AB      C2 A6 B9    JNZ APE1 ;NO, WAIT
11338    B9AE      3E 09 .     MVI A,STPRPT ;STOP RETURN FROM REPEATING
11339    B9B0      CD 08 48    CALL ZKBCTL
11340    B9B3      C3 43 00    JMP ZRSTDP ;RESTORE DISPLAY
11341    B9B6      . . .      ;
11342    B9B6      . . .      ;
11343    B9B6      82 41 55    APMSG1 DB IVON,'AUTO PLOT ERROR ',0
11344    B9CA      4D 41 58    APMSG2 DB 'MAXIMUM ',0
11345    B9D3      20 4C 45    APMSG3 DB ' LESS THAN OR EQUAL TO MINIMUM',ZEOP
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 304
11347	B9F2	.	.	*****	
11348	B9F2	.	.	; TICCHK--SEE IF USER HAS CALLED FOR MORE THAN	
11349	B9F2	.	.	; 1000 TICS	
11350	B9F2	.	.	; EXIT P => TOO MANY TICS	
11351	B9F2	.	.	; HL = POINTER TO X OR Y ERROR	
11352	B9F2	.	.	*****	
11353	B9F2	.	.	TICCHK EQU \$	
11354	B9F2	.	.	; CHECK X TICS FIRST	
11355	B9F2	.	.	; SEE IF ANY TICS AT ALL	
11356	B9F2	21	A8 FB	LXI H,XTICBF ;WAS A SPACING OF 0 SPECIFIE	
11357	B9F5	CD	6E BD	CALL LOD	
11358	B9F8	CD	59 BD	CALL TST	
11359	B9FB	CA	10 BA	JZ TCK010 ;YES, DONT CHECK COUNT	
11360	B9FE	21	B8 FB	LXI H,XMAXBF ;COMPUTE MAX-MIN	
11361	BA01	CD	6E BD	CALL LOD	
11362	BA04	21	BC FB	LXI H,XMINBF	
11363	BA07	CD	03 BD	CALL SB	
11364	BA0A	21	A8 FB	LXI H,XTICBF ;SEE HOW MANY TICS IN	
11365	BA0D	CD	84 BD	CALL DIV ;INTERVAL	
11366	BA10	CD	50 BD	CALL ABS	
11367	BA13	21	4A BA	LXI H,FP1001 ;SEE IF > 1000	
11368	BA16	CD	D3 BD	CALL SB	
11369	BA19	21	89 B9	LXI H,APXERR ;(ASSUME BAD)	
11370	BA1C	F0	.	RP ;YES, REPORT TOO MANY X TICS	
11371	BA1D	.	.	TCK010 EQU \$	
11372	BA1D	.	.	; CHECK Y TICS	
11373	BA1D	.	.	; SEE IF USER WANTS TICS	
11374	BA1D	21	A0 FB	LXI H,YTICBF ;LOAD ABS VAL OF SPACING	
11375	BA20	CD	6E BD	CALL LOD	
11376	BA23	CD	59 BD	CALL TST	
11377	BA26	CA	48 BA	JZ TCK020 ;NO Y TICS, DONT CHECK	
11378	BA29	21	80 FB	LXI H,YMAXBF ;COMPUTE MAX-MIN	
11379	BA2C	CD	6E BD	CALL LOD	
11380	BA2F	21	84 FB	LXI H,YMINBF	
11381	BA32	CD	D3 BD	CALL SB	
11382	BA35	21	A0 FB	LXI H,YTICBF ;SEE HOW MANY TICS	
11383	BA38	CD	B4 BD	CALL DIV	
11384	BA3B	CD	50 BD	CALL ABS	
11385	BA3E	21	4A BA	LXI H,FP1001 ;SEE IF > 1000	
11386	BA41	CD	D3 BD	CALL SB	
11387	BA44	21	8B B9	LXI H,APYERR ;(ASSUME BAD)	
11388	BA47	C9	.	RET ;P SET IF Y BAD	
11389	BA48	.	.	; RETURN FOR NO TICS WANTED	
11390	BA48	.	.	TCK020 EQU \$	
11391	BA48	3D	.	DCR A ;SET MINUS FLAG	
11392	BA49	C9	.	RET	
11393	BA4A	.	.	;	
11394	BA4A	8A	7A 40	FP1001 DB 212Q,172Q,100Q,0 ;1001 IN FLOAT PNT	
11395	BA4E	.	.	*****	
11396	BA4E	.	.	; TICERR--REPORT TOO MANY TICS	

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 305
=====
11397    BA4E      . . .      ; ENTRY HL = POINTER TO X OR Y MESSAGE
11398    BA4E      . . .      ;*****
11399    BA4E      . . .      TICERR EQU $
11400    BA4E      22 ED FF    SHLD ZMSGP3      ;SET MESSAGE POINTERS
11401    BA51      21 B6 B9    LXI H,APMSG1
11402    BA54      22 F1 FF    SHLD ZMSGP1
11403    BA57      21 66 BA    LXI H,TCMSG1
11404    BA5A      22 EF FF    SHLD ZMSGP2
11405    BA5D      21 70 BA    LXI H,TCMSG2
11406    BA60      22 EB FF    SHLD ZMSGP4
11407    BA63      C3 A2 B9    JMP APE010      ;DISPLAY MESSAGE
11408    BA66      . . .      ;
11409    BA66      54 4F 4F    TCMSG1 DB      'TOO MANY ',0
11410    BA70      20 54 49    TCMSG2 DB      ' TICS',ZEOP
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 306
11412	BA76	.	.	. ;*****	
11413	BA76	.	.	. ; LBLCHK--SEE IF MORE THAN 100 LABELS HAVE BEEN	
11414	BA76	.	.	. ; SPECIFIED	
11415	BA76	.	.	. ; EXIT P => TOO MANY LABELS	
11416	BA76	.	.	. ; HL = POINTER TO X OR Y MESSAGE	
11417	BA76	.	.	. ;*****	
11418	BA76	.	.	. LBLCHK EQU \$	
11419	BA76	.	.	. ; CHECK X LABELS FIRST	
11420	BA76	.	.	. ; SEE IF ANY AT ALL	
11421	BA76	21	AC	FB LXI H,XLBLBF ;0 TICS SPECIFIED?	
11422	BA79	CD	6E	BD CALL LOD	
11423	BA7C	CD	59	BD CALL TST	
11424	BA7F	CA	A1	BA JZ LCK010 ;YES, DONT CHECK FURTHER	
11425	BA82	21	B8	FB LXI H,XMAXBF ;COMPUTE MAX-MIN	
11426	BA85	CD	6E	BD CALL LOD	
11427	BA88	21	BC	FB LXI H,XMINBF	
11428	BA8B	CD	D3	BD CALL SB	
11429	BA8E	21	AC	FB LXI H,XLBLBF ;SEE HOW MANY LABELS	
11430	BA91	CD	B4	BD CALL DIV	
11431	BA94	CD	50	BD CALL ABS	
11432	BA97	21	CC	BA LXI H,FP101 ;SEE IF > 100	
11433	BA9A	CD	D3	BD CALL SB	
11434	BA9D	21	89	B9 LXI H,APXERR ;(ASSSUME BAD)	
11435	BAA0	F0	.	. RP ;RETURN IF TOU MANY	
11436	BAA1	.	.	. LCK010 EQU \$	
11437	BAA1	.	.	. ; CHECK Y LABELS	
11438	BAA1	21	A4	FB LXI H,YLBLBF ;SPACING OF 0 SPECIFIED?	
11439	BAA4	CD	6E	BD CALL LOD	
11440	BAA7	CD	59	BD CALL TST	
11441	BAAA	CA	48	BA JZ TCK020 ;YES, DONT CHECK FURTHER	
11442	BAAD	21	B0	FB LXI H,YMAXBF ;COMPUTE MAX-MIN	
11443	BAB0	CD	6E	BD CALL LOD	
11444	BAB3	21	B4	FB LXI H,YMINBF	
11445	BAB6	CD	D3	BD CALL SB	
11446	BAB9	21	A4	FB LXI H,YLBLBF ;SEE HOW MANY TICKS	
11447	BABC	CD	B4	BD CALL DIV	
11448	BABF	CD	50	BD CALL ABS	
11449	BAC2	21	CC	BA LXI H,FP101 ;SEE IF > 100	
11450	BAC5	CD	D3	BD CALL SB	
11451	BAC8	21	8B	B9 LXI H,APYERR ;(ASSUME BAD)	
11452	BACB	C9	.	. RET	
11453	BACC	.	.	. ;	
11454	BACC	87	4A	00 FP101 DB 207Q,112Q,0Q,0Q ;101 IN FLOATING PNT	

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 307
=====
11456    BAD0      . . .      ;*****
11457    BAD0      . . .      ; LBLERR--REPORT TOO MANY LABELS
11458    BAD0      . . .      ; ENTRY HL = POINTER TO X OR Y MESSAGE
11459    BAD0      . . .      ;*****
11460    BAD0      . . .      LBLERR EQU $
11461    BAD0      22 ED FF    SHLD ZMSGP3      ;SET MESSAGE POINTERS
11462    BAD3      21 B6 B9    LXI H,APMSG1
11463    BAD6      22 F1 FF    SHLD ZMSGP1
11464    BAD9      21 66 BA    LXI H,TCMSG1
11465    BADC      22 EF FF    SHLD ZMSGP2
11466    BADF      21 E8 BA    LXI H,LBMSG1
11467    BAE2      22 EB FF    SHLD ZMSGP4
11468    BAE5      C3 A2 B9    JMP APE010
11469    BAE8      . . .      ;
11470    BAE8      20 4C 41    LBMSG1 DB ' LABELS',ZEOP
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 308
11472	BAF0	. . .	;REF. NO. BC2	
11473	BAF0	. . .	;PROGRAM TITLE FLOATING POINT FORMAT CONVERSION PA	
11474	BAF0	. . .	;	
11475	BAF0	. . .	;	
11476	BAF0	. . .	;	
11477	BAF0	. . .	;	
11478	BAF0	. . .	;	
11479	BAF0	. . .	ORG 1354000 ;ORIGIN IS 46.75K	
11480	BB00	. . .	; 8008 BINARY FLOATING POINT SYSTEM	
11481	BB00	. . .	; FORMAT CONVERSION PACKAGE	
11482	BB00	. . .	; PROGRAMMER CAL OHME	
11483	BB00	. . .	; DATE 26 DECEMBER 1973	
11484	BB00	. . .	; ARITH IS THE BEGINNING ADDRESS OF THE	
11485	BB00	. . .	; ARITHMETIC AND UTILITY PACKAGE OF THE FLOATI	
11486	BB00	. . .	; POINT SYSTEM.	
11487	BB00	. . .	; SCR IS THE BEGINNING ADDRESS OF THE	
11488	BB00	. . .	; RAM USED AS SCRATCPAD FOR THE SYSTEM.	
11489	BB00	. . .	; RAM LOCATIONS USED BY THE BINARY	
11490	BB00	. . .	; FLOATING POINT SYSTEM.	
11491	BB00	. . .	; SUBROUTINE TO CONVERT FROM FIXED	
11492	BB00	. . .	; POINT TO FLOATING POINT FORMAT.	
11493	BB00	6B . .	FLT MOV L,E ;INPUT EXPONENT	
11494	BB01	5A . .	MOV E,D ;4TH INPUT FRACTION	
11495	BB02	51 . .	MOV D,C ;3RD INPUT FRACTION	
11496	BB03	48 . .	MOV C,B ;2ND INPUT FRACTION	
11497	BB04	47 . .	MOV B,A ;1ST INPUT FRACTION	
11498	BB05	7D . .	MOV A,L ;INPUT EXPONENT	
11499	BB06	EE 80 .	XRI 2000 ;APPLY EXPONENT BIAS	
11500	BB08	26 90 .	MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
11501	BB0A	2E 30 .	MVI L,ACCE ;TO ADDR ACCUM EXPONENT	
11502	BB0C	77 . .	MOV M,A ;ACCUMULATOR EXPONENT	
11503	BB0D	2C . .	INR L ;TO ADDRESS ACCUM SIGN	
11504	BB0E	36 80 .	MVI M,2000 ;SET ACCUM SIGN POSITIVE	
11505	BB10	2C . .	INR L ;TO ADDR ACCUM 1ST FRCTN	
11506	BB11	78 . .	MOV A,B ;1ST INPUT FRACTION	
11507	BB12	A7 . .	ANA A ;SET SIGN BIT	
11508	BB13	17 . .	RAL ;INPUT SIGN TO CARRY	
11509	BB14	C3 6A BE	JMP ADD10 ;COMPLETE CONVERSION	
11510	BB17	. . .	; SUBROUTINE TO CONVERT FROM FLOATING	
11511	BB17	. . .	; POINT TO FIXED POINT FORMAT.	
11512	BB17	26 90 .	FIX MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
11513	BB19	2E 30 .	MVI L,ACCE ;TO ADDR ACCUM EXPONENT	
11514	BB1B	7E . .	MOV A,M ;ACCUMULATOR EXPONENT	
11515	BB1C	A7 . .	ANA A ;SET CONTROL BITS	
11516	BB1D	CA 45 BB	JZ FIX1 ;IF ACCUMULATOR IS ZERO	
11517	BB20	7B . .	MOV A,E ;INPUT EXPONENT	
11518	BB21	C6 7F .	ADI 1770 ;APPLY BIAS - 1	
11519	BB23	96 . .	SUB M ;SHIFT COUNT - 1	
11520	BB24	D8 . .	RC ;RETURN IF ACCUM TOO LARGE	
11521	BB25	FE 1F .	CPI 0370 ;COMPARE TO LARGE SHIFT	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS
11522	BB27	D2	45	BB	JNC FIX1 ;IF ACCUMULATOR TOO SMALL
11523	BB2A	C6	01	.	ADI 1 ;SHIFT COUNT
11524	BB2C	2E	32	.	MVI L,ACC1 ;TO ADDR ACCUM 1ST FRCTN
11525	BB2E	46	.	.	MOV B,M ;ACCUMULATOR 1ST FRACTIO
11526	BB2F	2C	.	.	INR L ;TO ADDR ACCUM 2ND FRCTN
11527	BB30	4E	.	.	MOV C,M ;ACCUMULATOR 2ND FRCTN
11528	BB31	2C	.	.	INR L ;TO ADDR ACCUM 3RD FRCTN
11529	BB32	56	.	.	MOV D,M ;ACCUMULATOR 3RD FRCTN
11530	BB33	CD	C7	BE	CALL RSH ;POSITION THE FRACTION
11531	BB36	2E	31	.	MVI L,ACCS ;TO ADDR ACCUM SIGN
11532	BB38	7E	.	.	MOV A,M ;ACCUMULATOR SIGN
11533	BB39	A7	.	.	ANA A ;SET CONTROL BITS
11534	BB3A	F4	ED	BE	CP COMP ;COMPLEMENT FRCTN IF NEG
11535	BB3D	3E	01	.	MVI A,1 ;NON-ZERO
11536	BB3F	B0	.	.	ORA B ;SET CONTROL BITS FOR EXIT
11537	BB40	78	.	.	MOV A,B ;1ST RESULT
11538	BB41	41	.	.	MOV B,C ;2ND RESULT
11539	BB42	4A	.	.	MOV C,D ;3RD RESULT
11540	BB43	53	.	.	MOV D,E ;4TH RESULT
11541	BB44	C9	.	.	RET ;RETURN TO CALLER
11542	BB45	AF	.	.	FIX1 XRA A ;ZERO
11543	BB46	47	.	.	MOV B,A ;ZERO
11544	BB47	4F	.	.	MOV C,A ;ZERO
11545	BB48	57	.	.	MOV D,A ;ZERO
11546	BB49	C9	.	.	RET ;RETURN TO CALLER
11547	BB4A	.	.	.	; INP SUBROUTINE ENTRY POINT.
11548	BB4A	.	.	.	; INITIALIZE TEMPORARY STORAGE.
11549	BB4A	5E	.	.	INP MOV E,M ;FIRST CHARACTER OF STRI
11550	BB4B	CD	DE	BC	CALL SVAD ;SET CHAR ADDR, PNT FLG, EXP
11551	BB4E	2C	.	.	INR L ;TO ADDRESS VALUE SIGN
11552	BB4F	36	80	.	MVI M,200Q ;SET VALUE SIGN POSITIVE
11553	BB51	2E	30	.	MVI L,ACCE ;TO ADDR ACCUM EXPONENT
11554	BB53	72	.	.	MOV M,D ;SET ACCUM TO ZERO
11555	BB54	7B	.	.	MOV A,E ;FIRST CHARACTER
11556	BB55	.	.	.	;*****
11557	BB55	D6	30	.	SUI 60Q ;CONVERT TO DIGIT
11558	BB57	.	.	.	;*****
11559	BB57	FE	F0	.	CPI 360Q ;COMPARE TO SPACE
11560	BB59	CA	69	BB	JZ INP1 ;IF SPACE CHARACTER
11561	BB5C	FE	FB	.	CPI 373Q ;COMPARE CHAR TO PLUS
11562	BB5E	CA	69	BB	JZ INP1 ;IF PLUS SIGN
11563	BB61	FE	FD	.	CPI 375Q ;COMPARE TO MINUS
11564	BB63	C2	71	BB	JNZ INP2 ;IF NOT MINUS SIGN
11565	BB66	2E	3A	.	MVI L,TMP3 ;TO ADDR VALUE SIGN
11566	BB68	72	.	.	MOV M,D ;SET VALUE SIGN NEGATIVE
11567	BB69	.	.	.	; ANALYZE NEXT CHARACTER IN STRING.
11568	BB69	CD	EB	BC	INP1 CALL CHAD ;CALL CHAR ADDR SBRTN
11569	BB6C	7E	.	.	MOV A,M ;NEXT CHARACTER
11570	BB6D	.	.	.	;*****
11571	BB6D	D6	30	.	SUI 60Q ;CONVERT TO DIGIT

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 310
=====
11572    BB6F      .      .      .      ;*****
11573    BB6F      26  90  .      MVI      H,SCRB      ;TO ADDRESS SCRATCH BANK
11574    BB71      06  00  .      INP2     MVI      B,0        ;DIGIT 2ND WD OR DEC EXP
11575    BB73      FE  FE  .      CPI      3760       ;COMPARE TO DECIMAL POINT
11576    BB75      CA  AE  BB     JZ      INP3       ;IF DECIMAL POINT
11577    BB78      FE  15  .      CPI      0250       ;COMPARE TO EXPONENT SIGN
11578    BB7A      CA  B8  BB     JZ      INP4       ;IF EXPONENT SIGN
11579    BB7D      FE  0A  .      CPI      120        ;SET CARRY IF CHAR IS DIGIT
11580    BB7F      D2  EF  BB     JNC     INP8       ;IF CHAR IS NOT A DIGIT
11581    BB82      2E  3F  .      MVI      L,TMP4     ;TO ADDR CURRENT DIGIT
11582    BB84      77  .   .      MOV      M,A        ;SAVE CURRENT DIGIT
11583    BB85      21  F6  BC     LXI      H,FTEN     ;TO ADDR FLOATING TEN
11584    BB88      CD  8C  BD     CALL     MUL        ;MULTIPLY BY TEN
11585    BB8B      2E  3B  .      MVI      L,VALE     ;TO ADDR VALUE
11586    BB8D      CD  3E  BD     CALL     STR        ;STORE OLD VALUE TIMES TEN
11587    BB90      2C  .   .      INR      L          ;TO ADDR CURRENT DIGIT
11588    BB91      7E  .   .      MOV      A,M        ;CURRENT DIGIT
11589    BB92      06  00  .      MVI      B,0        ;CLEAR 2ND WORD OF DIGIT
11590    BB94      48  .   .      MOV      C,B        ;CLEAR 3RD WORD OF DIGIT
11591    BB95      50  .   .      MOV      D,B        ;CLEAR 4TH WORD OF DIGIT
11592    BB96      1E  08  .      MVI      E,0100     ;INDICATE DIGIT IS IN REG
11593    BB98      CD  00  BB     CALL     FLT        ;CONVERT DIGIT TO FLOATING P
11594    BB9B      2E  3B  .      MVI      L,VALE     ;TO ADDR VALUE
11595    BB9D      CD  D6  BD     CALL     AD         ;ADD OLD VALUE TIMES TEN
11596    BBA0      2E  39  .      MVI      L,TMP2     ;TO ADDR DEC PNT FLAG
11597    BBA2      7E  .   .      MOV      A,M        ;DECIMAL POINT FLAG
11598    BBA3      A7  .   .      ANA      A          ;SET CONTROL BITS
11599    BBA4      CA  69  BB     JZ      INP1       ;IF NO DEC PNT ENCOUNTERED
11600    BBA7      2D  .   .      DCR      L          ;TO ADDR INPUT EXPONENT
11601    BBA8      46  .   .      MOV      B,M        ;INPUT EXPONENT
11602    BBA9      05  .   .      DCR      B          ;DECREMENT INPUT EXPONENT
11603    BBAA      70  .   .      MOV      M,B        ;UPDATE INPUT EXPONENT
11604    BBAB      C3  69  BB     JMP      INP1       ;TO GET NEXT CHARACTER
11605    BBAE      2E  39  .      INP3     MVI      L,TMP2     ;TO ADDR DEC PNT FLAG
11606    BBB0      AE  .   .      XRA      M          ;ZERO IF FLAG SET
11607    BBB1      77  .   .      MOV      M,A        ;SET DEC PNT FLAG
11608    BBB2      C2  69  BB     JNZ     INP1       ;IF FLAG NOT ALREADY SET
11609    BBB5      C3  EF  BB     JMP      INP8       ;IF 2ND DEC PNT
11610    BBB8      .   .   .      ;      PROCESS DECIMAL EXPONENT.
11611    BBB8      CD  EB  BC     INP4     CALL     CHAD     ;CALL CHAR ADDR SBRTN
11612    BBBB      7E  .   .      MOV      A,M        ;NEXT CHARACTER OF STRIN
11613    BBBC      .   .   .      ;*****
11614    BBBC      D6  30  .      SUI      60Q       ;CONVERT TO DIGIT
11615    BBBE      .   .   .      ;*****
11616    BBBE      47  .   .      MOV      B,A        ;CURRENT CHARACTER
11617    BBBF      D6  FD  .      SUI      375Q     ;COMPARE TO MINUS CHAR
11618    BBC1      5F  .   .      MOV      E,A        ;CHAR - MINUS SIGN
11619    BBC2      CA  CB  BB     JZ      INP5       ;IF MINUS SIGN
11620    BBC5      C6  02  .      ADI      2         ;COMPARE TO PLUS CHAR
11621    BBC7      78  .   .      MOV      A,B        ;CURRENT CHARACTER
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 311
11622	BBC8	C2 CF BB	JNZ INP6 ;IF NOT PLUS SIGN	
11623	BBC8	. . .	INP5 EQU \$	
11624	BBC8	2C . .	INR L ;TO ADDRESS NEXT CHAR	
11625	BBC8	7E . .	MOV A,M ;NEXT CHARACTER OF STRIN	
11626	BBCD	. . .	*****	
11627	BBCD	06 30 .	SUI 60Q ;CONVERT TO DIGIT	
11628	BBCF	. . .	*****	
11629	BBCF	06 00 .	INP6 MVI B,0 ;POSSIBLE DEC EXPONENT	
11630	BBD1	FE 0A .	CPI 12Q ;SET CARRY IF CHAR IS DIGIT	
11631	BBD3	02 EF BB	JNC INP8 ;IF CHAR IS NOT A DIGIT	
11632	BBD6	47 . .	MOV B,A ;DEC EXP EQUAL DIGIT	
11633	BBD7	2C . .	INR L ;TO ADDRESS NEXT CHAR	
11634	BBD8	7E . .	MOV A,M ;NEXT CHARACTER OF STRIN	
11635	BBD9	. . .	*****	
11636	BBD9	D6 30 .	SUI 60Q ;CONVERT TO DIGIT	
11637	BBDB	. . .	*****	
11638	BBDB	FE 0A .	CPI 12Q ;SET CARRY IF CHAR IS DIGIT	
11639	BBDD	D2 E8 BB	JNC INP7 ;IF CHAR IS NOT A DIGIT	
11640	BBE0	. . .	; FORM COMPLETE DECIMAL EXPONENT.	
11641	BBE0	4F . .	MOV C,A ;LS DIGIT OF DEC EXP	
11642	BBE1	78 . .	MOV A,B ;MS DIGIT OF DEC EXP	
11643	BBE2	87 . .	ADD A ;2 * MS DIGIT	
11644	BBE3	87 . .	ADD A ;4 * MS DIGIT	
11645	BBE4	80 . .	ADD B ;5 * MS DIGIT	
11646	BBE5	87 . .	ADD A ;10 * MS DIGIT	
11647	BBE6	81 . .	ADD C ;10 +* MS + LS DIGIT	
11648	BBE7	47 . .	MOV B,A ;B = DECIMAL EXPONENT	
11649	BBE8	7B . .	INP7 MOV A,E ;SIGN OF DEC EXPONENT	
11650	BBE9	A7 . .	ANA A ;SET CONTROL BITS	
11651	BBEA	C2 EF BB	JNZ INP8 ;IF SIGN PLUS	
11652	BBED	90 . .	SUB B ;COMPLEMENT DEC EXP	
11653	BBEF	47 . .	MOV B,A ;DECIMAL EXPONENT	
11654	BBEF	26 90 .	INP8 MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
11655	BBF1	2E 3A .	MVI L,TMP3 ;TO ADDRESS INPUT SIGN	
11656	BBF3	4E . .	MOV C,M ;INPUT SIGN	
11657	BBF4	2E 31 .	MVI L,ACCS ;TO ADDRESS ACCUM SIGN	
11658	BBF6	71 . .	MOV M,C ;ACCUMULATOR SIGN	
11659	BBF7	78 . .	MOV A,B ;DECIMAL EXPONENT	
11660	BBF8	. . .	; CONVERT DECIMAL EXPONENT TO BINARY.	
11661	BBF8	2E 38 .	INP9 MVI L,TMP1 ;TO ADDRESS DEC EXPONENT	
11662	BBFA	86 . .	ADD M ;ADJUST DECIMAL EXPONENT	
11663	BBFB	CA 59 BD	JZ TST ;IN DEC EXP IS ZERO	
11664	BBFE	77 . .	MOV M,A ;CURRENT DECIMAL EXPONEN	
11665	BBFF	21 F6 BC	LXI H,FTEN ;TO ADDR FLOATING TEN	
11666	BC02	F2 0D BC	JP INP10 ;IF MULTIPLY REQUIRED	
11667	BC05	CD B4 BD	CALL DIV ;DIVIDE BY TEN	
11668	BC08	3E 01 .	MVI A,1 ;TO INCREMENT DEC EXP	
11669	BC0A	C3 F8 BB	JMP INP9 ;TO TEST FOR COMPLETION	
11670	BC0D	CD 8C BD	INP10 CALL MUL ;MULTIPLY BY TEN	
11671	BC10	D8 . .	RC ;RETURN IF OVERFLOW	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 312
11672	BC11	3E	FF	.	MVI A,377Q ;TO DECREMENT DEC EXP	
11673	BC13	C3	F8	BB	JMP INP9 ;TO TEST FOR COMPLETION	
11674	BC16	.	.	.	; OUT SUBROUTINE ENTRY POINT.	
11675	BC16	.	.	.	; SAVE CHARACTER ADDRESS AND ACCUMULATOR.	
11676	BC16	2D	.	.	OU DCR L ;DECREMENT CHARACTER ADDRE	
11677	BC17	CD	DE	BC	CALL SVAD ;SET CHAR ADDR, DIG CNT, DEC	
11678	BC1A	CD	59	BD	CALL TST ;LOAD ACCUM TO REGISTERS	
11679	BC1D	2E	38	.	MVI L,VALE ;TO ADDR ACCUM SAVE AREA	
11680	BC1F	CD	3E	BD	CALL STR ;CALL REG STR SUBROUTINE	
11681	BC22	.	.	.	; OUTPUT SIGN CHARACTER.	
11682	BC22	CD	EB	BC	CALL CHAD ;GET CHAR ADDRESS	
11683	BC25	36	F0	.	MVI M,360Q ;STORE SPACE CHARACTER	
11684	BC27	A7	.	.	ANA A ;SET CONTROL BITS	
11685	BC28	CA	44	BC	JZ OUT3 ;IF ACCUMULATOR IS ZERO	
11686	BC2B	5F	.	.	MOV E,A ;ACCUMULATOR EXPONENT	
11687	BC2C	78	.	.	MOV A,B ;ACCUM SIGN AND 1ST FRCT	
11688	BC2D	A7	.	.	ANA A ;SET CONTROL BITS	
11689	BC2E	7B	.	.	MOV A,E ;ACCUMULATOR EXPONENT	
11690	BC2F	F2	34	BC	JP OUT1 ;IF ACCUM IS POSITIVE	
11691	BC32	36	FD	.	MVI M,375Q ;CHANGE SIGN TO MINUS	
11692	BC34	.	.	.	; SCALE ACCUMULATOR TO .1 - 1. RANGE.	
11693	BC34	FE	7E	.	OUT1 CPI 176Q ;COMPARE TO SMALL EXPONENT	
11694	BC36	21	F6	BC	OUT2 LXI H,FTEN ;TO ADDR FLOATING TEN	
11695	BC39	DA	4E	BC	JC OUT4 ;IF EXPONENT TOO SMALL	
11696	BC3C	FE	81	.	CPI 201Q ;COMPARE TO LARGE EXP	
11697	BC3E	DA	59	BC	JC OUT5 ;IF EXP SMALL ENOUGH	
11698	BC41	CD	84	BD	CALL DIV ;DIVIDE BY TEN	
11699	BC44	26	90	.	OUT3 MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
11700	BC46	2E	39	.	MVI L,TMP2 ;TO ADDR DECIMAL EXPONENT	
11701	BC48	5E	.	.	MOV E,M ;DECIMAL EXPONENT	
11702	BC49	1C	.	.	INR E ;INCREMENT DECIMAL EXPONEN	
11703	BC4A	73	.	.	MOV M,E ;DECIMAL EXPONENT	
11704	BC4B	C3	36	BC	JMP OUT2 ;TO TEST FOR SCALING COMPLET	
11705	BC4E	CD	8C	BD	OUT4 CALL MUL ;MULTIPLY BY TEN	
11706	BC51	2E	39	.	MVI L,TMP2 ;TO ADDR DECIMAL EXPONENT	
11707	BC53	5E	.	.	MOV E,M ;DECIMAL EXPONENT	
11708	BC54	1D	.	.	DCR E ;DECREMENT DECIMAL EXPONEN	
11709	BC55	73	.	.	MOV M,E ;DECIMAL EXPONENT	
11710	BC56	C3	34	BC	JMP OUT1 ;TO TEST FOR SCALING COMPLET	
11711	BC59	.	.	.	; ROUND THE VALUE BY ADDING .00000005.	
11712	BC59	CD	50	BD	OUT5 CALL ABS ;SET ACCUM POSITIVE	
11713	BC5C	21	FA	BC	LXI H,RND0 ;ADD .00000005	
11714	BC5F	CD	D6	BD	CALL AD ;ADD THE ROUNDER	
11715	BC62	FE	81	.	CPI 201Q ;CHECK FOR OVERFLOW	
11716	BC64	D2	36	BC	JNC OUT2 ;IF EXP TOO LARGE	
11717	BC67	.	.	.	; SET DIGIT COUNTS.	
11718	BC67	2E	39	.	MVI L,TMP2 ;TO ADDR DECIMAL EXPONENT	
11719	BC69	7E	.	.	MOV A,M ;DECIMAL EXPONENT	
11720	BC6A	5F	.	.	MOV E,A ;DIGITS BEFORE DEC POINT	
11721	BC6B	FE	08	.	CPI 010Q ;COMPARE TO LARGE EXP	

=====					PAGE 313		
ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS				
=====							
11722	BC6D	DA 72 BC		JC	OUT6	;IF EXPONENT IN RANGE	
11723	BC70	1E 01 .		MVI	E,1	;DIGITS BEFORE DEC POINT	
11724	BC72	93 . .	OUT6	SUB	E	;ADJUST DEC EXPONENT	
11725	BC73	77 . .		MOV	M,A	;DECIMAL EXPONENT	
11726	BC74	3E 07 .		MVI	A,7	;TOTAL NUMBER OF DIGITS	
11727	BC76	93 . .		SUB	E	;DIGITS AFTER DECIMAL PNT	
11728	BC77	2C . .		INR	L	;TO ADDR 2ND DIGIT CNT	
11729	BC78	77 . .		MOV	M,A	;DIGITS AFTER DECIMAL PO	
11730	BC79	1D . .		DCR	E	;DECREMENT DIGIT COUNT	
11731	BC7A	7B . .		MOV	A,E	;DIGITS BEFORE DEC PNT	
11732	BC7B	. . .		; OUTPUT SIGNIFICANT DIGITS.			
11733	BC7B	2E 38 .	OUT7	MVI	L,TMP1	;TO ADDR DIGIT COUNT	
11734	BC7D	86 . .		ADD	M	;ADJUST DIGIT COUNT	
11735	BC7E	77 . .		MOV	M,A	;NEW DIGIT COUNT	
11736	BC7F	FA 9C BC		JM	OUT8	;IF COUNT RUN OUT	
11737	BC82	21 F6 BC		LXI	H,FTEN	;TO ADDR FLOATING TEN	
11738	BC85	CD 8C BD		CALL	MUL	;MULTIPLY BY TEN	
11739	BC88	1E 08 .		MVI	E,100	;BINARY SCALING FACTOR	
11740	BC8A	CD 17 BB		CALL	FIX	;CONVERT TO FIXED FORMAT	
11741	BC8D	CD EB BC		CALL	CHAD	;GET CHARACTER ADDRESS	
11742	BC90	77 . .		MOV	M,A	;OUTPUT DECIMAL DIGIT	
11743	BC91	AF . .		XRA	A	;CLEAR CURRENT DIGIT	
11744	BC92	1E 08 .		MVI	E,0100	;BINARY SCALING FACTOR	
11745	BC94	CD 00 BB		CALL	FLT	;RESTORE VALUE MINUS DIGIT	
11746	BC97	3E FF .		MVI	A,3770	;TO ADJUST DIGIT CNT	
11747	BC99	C3 7B BC		JMP	OUT7	;LOOP FOR NEXT DIGIT	
11748	BC9C	2E 3A .	OUT8	MVI	L,TMP3	;TO ADDR 2ND DIGIT CNT	
11749	BC9E	7E . .		MOV	A,M	;DIGITS AFTER DECIMAL PN	
11750	BC9F	36 FF .		MVI	M,3770	;SET 2ND COUNT NEG	
11751	BCA1	A7 . .		ANA	A	;SET CONTROL BITS	
11752	BCA2	FA AF BC		JM	OUT9	;IF 2ND COUNT RAN OUT	
11753	BCA5	CD EB BC		CALL	CHAD	;GET CHARACTER ADDRESS	
11754	BCA8	36 FE .		MVI	M,3760	;STORE DECIMAL POINT	
11755	BCAA	26 90 .		MVI	H,SCRB	;TO ADDRESS SCRATCH BANK	
11756	BCAC	C3 7B BC		JMP	OUT7	;LOOP FOR NEXT DIGIT	
11757	BCAF	2D . .	OUT9	DCR	L	;TO ADDR DECIMAL EXP	
11758	BCB0	A6 . .		ANA	M	;DECIMAL EXPONENT	
11759	BCB1	CA D6 BC		JZ	OUT13	;IF DECIMAL EXPONENT IS ZERO	
11760	BCB4	. . .		; OUTPUT DECIMAL EXPONENT.			
11761	BCB4	06 FB .		MVI	B,3730	;PLUS CHARACTER	
11762	BCB6	F2 BE BC		JP	OUT10	;IF EXPONENT IS POSITIVE	
11763	BCB9	06 FD .		MVI	B,3750	;CHANGE SIGN TO MINUS	
11764	BCBB	4F . .		MOV	C,A	;NEGATIVE EXPONENT	
11765	BCBC	AF . .		XRA	A	;ZERO	
11766	BCBD	91 . .		SUB	C	;COMPLEMENT EXPONENT	
11767	BCBE	0E FF .	OUT10	MVI	C,3770	;EMBRYO TENS DIGIT	
11768	BCC0	57 . .	OUT11	MOV	D,A	;UNITS DIGIT	
11769	BCC1	0C . .		INR	C	;INCREMENT TENS DIGIT	
11770	BCC2	D6 0A .		SUI	0120	;REDUCE REMAINDER	
11771	BCC4	D2 C0 BC		JNC	OUT11	;IF MORE TENS	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 314
11772	BCC7	3E	15	.	MVI A,025Q ;EXPONENT SIGN	
11773	BCC9	.	.	.	OUT12 EQU \$	
11774	BCC9	CD	EB	BC	CALL CHAD ;GET CHAR ADDRESS	
11775	BCCC	CD	3E	BD	CALL STR ;STORE LAST 4 CHARACTERS	
11776	BCCF	26	90	.	MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
11777	BCD1	2E	3B	.	MVI L,VALE ;TO ADDRESS ACCUM SAVE ARE	
11778	BCD3	C3	6E	BD	JMP LOD ;RESTORE ACCUM AND EXIT	
11779	BCD6	.	.	.	; OUTPUT 4 SPACES IF EXPONENT IS ZERO.	
11780	BCD6	3E	F0	.	OUT13 MVI A,360Q ;SPACE CHARACTER	
11781	BCD8	47	.	.	MOV B,A ;SPACE CHARACTER	
11782	BCD9	4F	.	.	MOV C,A ;SPACE CHARACTER	
11783	BCDA	57	.	.	MOV D,A ;SPACE CHARACTER	
11784	BCDB	C3	C9	BC	JMP OUT12 ;TO STORE CHARACTERS	
11785	BCDE	.	.	.	; SUBROUTINE TO SAVE CHARACTER STRING ADDR.	
11786	BCDE	7D	.	.	SVAD MOV A,L ;CHARACTER STRING WORD	
11787	BCDF	44	.	.	MOV B,H ;CHARACTER STRING BANK	
11788	BCE0	0E	00	.	MVI C,0 ;INPUT EXP OR DIGIT CNT	
11789	BCE2	51	.	.	MOV D,C ;DEC PNT FLAG OR DEC EXP	
11790	BCE3	26	90	.	MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
11791	BCE5	2E	36	.	MVI L,ADRL ;TO ADDR CHAR STRING WORD	
11792	BCE7	CD	3E	BD	CALL STR ;STORE A, B, C, AND D	
11793	BCEA	C9	.	.	RET ;RETURN TO CALLER	
11794	BCEB	.	.	.	; SUBROUTINE TO OBTAIN NEXT CHARACTER ADDR.	
11795	BCEB	26	90	.	CHAD MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
11796	BCED	2E	36	.	MVI L,ADRL ;TO ADDR CHAR STRING WORD	
11797	BCEF	5E	.	.	MOV E,M ;CHARACTER STRING WORD	
11798	BCF0	1C	.	.	INR E ;TO ADDRESS NEXT CHAR	
11799	BCF1	73	.	.	MOV M,E ;UPDATE CHAR STRING WORD	
11800	BCF2	2C	.	.	INR L ;TO ADDR CHAR STRING BANK	
11801	BCF3	66	.	.	MOV H,M ;CHARACTER STRING BANK	
11802	BCF4	6B	.	.	MOV L,E ;CHARACTER STRING WORD	
11803	BCF5	C9	.	.	RET ;RETURN TO CALLER	
11804	BCF6	84	20	00	FTEN DB 204Q,040Q,0,0 ;FLOATING TEN	
11805	BCFA	68	56	BF	RNDU DB 150Q,126Q,277Q,255Q ;.00000005	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 315
11807	BCFE	. . .	;REF. NO. BC1	
11808	BCFE	. . .	;PROGRAM TITLE FLOATING POINT MATH PACKAGE	
11809	BCFE	. . .	;	
11810	BCFE	. . .	;	
11811	BCFE	. . .	;	
11812	BCFE	. . .	;	
11813	BCFE	. . .	;	
11814	BCFE	. . .	ORG 1364000	
11815	BD00	. . .	ARITH EQU \$	
11816	00BD	. . .	ARTHB EQU ARITH/256 ;BANK NO.	
11817	BD00	. . .	; 8008 BINARY FLOATING POINT SYSTEM	
11818	BD00	. . .	; ARITHMETIC AND UTILITY PACKAGE	
11819	BD00	. . .	; PROGRAMMER CAL OHME	
11820	BD00	. . .	; DATE 26 DECEMBER 1973	
11821	BD00	. . .	; ARITH IS THE BEGINNING ADDRESS OF THE	
11822	BD00	. . .	; ARITHMETIC AND UTILITY PACKAGE OF THE FLOATI	
11823	BD00	. . .	; POINT SYSTEM.	
11824	BD00	. . .	; SCR IS THE BEGINNING ADDRESS OF THE	
11825	BD00	. . .	; RAM USED AS SCRATCPAD FOR THE SYSTEM.	
11826	BD00	. . .	; THE RAM MULTIPLY AND DIVIDE SUBROUTINES	
11827	BD00	. . .	; ARE MOVED FROM ROM TO RAM BY SUBROUTINE	
11828	BD00	. . .	; INIT AND ARE EXECUTED IN RAM ONLY.	
11829	BD00	. . .	; RAM MULTIPLY SUBROUTINE.	
11830	9000	. . .	MULX4 EQU \$-ARITH+SCR	
11831	BD00	C6 00 .	ADI 0 ;ADD OPERAND 3RD FRACTION	
11832	0001	. . .	MULP3 EQU \$-1-ARITH	
11833	BD02	5F . . .	MOV E,A ;4TH PARTIAL PRODUCT	
11834	BD03	7A . . .	MOV A,D ;3RD PARTIAL PRODUCT	
11835	BD04	CE 00 .	ACI 0 ;ADD OPERAND 2ND FRACTION	
11836	0005	. . .	MULP2 EQU \$-1-ARITH	
11837	BD06	57 . . .	MOV D,A ;3RD PARTIAL PRODUCT	
11838	BD07	79 . . .	MOV A,C ;2ND PARTIAL PRODUCT	
11839	BD08	CE 00 .	ACI 0 ;ADD OPERAND 1ST FRACTION	
11840	0009	. . .	MULP1 EQU \$-1-ARITH	
11841	BD0A	C3 71 BF	JMP MULX5 ;TO ROM CODE	
11842	BD0D	. . .	; RAM DIVIDE SUBROUTINE.	
11843	900D	. . .	DIVX5 EQU \$-ARITH+SCR	
11844	BD0D	D6 00 .	SUI 0 ;SUB DIVISOR 4TH FRACTION	
11845	000E	. . .	OP4S EQU \$-1-ARITH	
11846	BD0F	7D . . .	MOV A,L ;REMAINDER 3RD FRACTION	
11847	BD10	DE 00 .	SBI 0 ;SUB DIVISOR 3RD FRACTION	
11848	0011	. . .	OP3S EQU \$-1-ARITH	
11849	BD12	6F . . .	MOV L,A ;REMAINDER 3RD FRACTION	
11850	BD13	7C . . .	MOV A,H ;REMAINDER 2ND FRACTION	
11851	BD14	DE 00 .	SBI 0 ;SUB DIVISOR 2ND FRACTION	
11852	0015	. . .	OP2S EQU \$-1-ARITH	
11853	BD16	67 . . .	MOV H,A ;REMAINDER 2ND FRACTION	
11854	BD17	7B . . .	MOV A,E ;REMAINDER 1ST FRACTION	
11855	BD18	DE 00 .	SBI 0 ;SUB DIVISOR 1ST FRACTION	
11856	0019	. . .	OP1S EQU \$-1-ARITH	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 316
11857	BD1A	SF	.	.	MOV E,A ;REMAINDER 1ST FRACTION	
11858	BD1B	3E	00	.	MVI A,0 ;REMAINDER 4TH FRACTION	
11859	001C	.	.	OP4A	EQU \$-1-ARITH	
11860	BD1D	C9	.	.	RET ;RETURN TO ROM	
11861	901E	.	.	DIVX6	EQU \$-ARITH+SCR	
11862	BD1E	C6	00	.	ADI 0 ;ADD DIVISOR 3RD FRACTION	
11863	001F	.	.	OP3A	EQU \$-1-ARITH	
11864	BD20	6F	.	.	MOV L,A ;REMAINDER 3RD FRACTION	
11865	BD21	7C	.	.	MOV A,H ;REMAINDER 2ND FRACTION	
11866	BD22	CE	00	.	ACI 0 ;ADD DIVISOR 2ND FRACTION	
11867	0023	.	.	OP2A	EQU \$-1-ARITH	
11868	BD24	67	.	.	MOV H,A ;REMAINDER 2ND FRACTION	
11869	BD25	7B	.	.	MOV A,E ;REMAINDER 1ST FRACTION	
11870	BD26	CE	00	.	ACI 0 ;ADD DIVISOR 1ST FRACTION	
11871	0027	.	.	OP1A	EQU \$-1-ARITH	
11872	BD28	SF	.	.	MOV E,A ;REMAINDER 1ST FRACTION	
11873	BD29	3E	00	.	MVI A,0 ;REMAINDER 4TH FRACTION	
11874	002A	.	.	OP4X	EQU \$-1-ARITH	
11875	BD2B	C3	DD	BF	JMP DIVX2 ;TO ROM CODE	
11876	BD2E	.	.	.	; RAM LOCATIONS USED BY THE BINARY	
11877	BD2E	.	.	.	; FLOATING POINT SYSTEM.	
11878	002E	.	.	OVER	EQU \$-ARITH	
11879	BD2E	00	.	.	DB 0 ;INITIALLY CLEAR	
11880	002F	.	.	PREX	EQU OVER+1 ;PREVIOUS EXPONENT	
11881	0030	.	.	ACCE	EQU PREX+1 ;ACCUMULATOR EXPONENT	
11882	0031	.	.	ACCS	EQU ACCE+1 ;ACCUMULATOR SIGN	
11883	0032	.	.	ACC1	EQU ACCS+1 ;ACCUMULATOR 1ST FRACTION	
11884	0033	.	.	ACC2	EQU ACC1+1 ;ACCUMULATOR 2ND FRACTION	
11885	0034	.	.	ACC3	EQU ACC2+1 ;ACCUMULATOR 3RD FRACTION	
11886	0035	.	.	SF	EQU ACC3+1 ;SUBTRACTION FLAG	
11887	0036	.	.	ADRL	EQU SF+1 ;CHARACTER STRING WORD	
11888	0037	.	.	ADRH	EQU ADRL+1 ;CHARACTER WORD BANK	
11889	0038	.	.	TMP1	EQU ADRH+1 ;TEMPORARY STORAGE	
11890	0039	.	.	TMP2	EQU TMP1+1 ;TEMPORARY STORAGE	
11891	003A	.	.	TMP3	EQU TMP2+1 ;TEMPORARY STORAGE	
11892	003B	.	.	VALE	EQU TMP3+1 ;VALUE EXPONENT	
11893	003C	.	.	VAL1	EQU VALE+1 ;VALUE 1ST FRACTION	
11894	003D	.	.	VAL2	EQU VAL1+1 ;VALUE 2ND FRACTION	
11895	003E	.	.	VAL3	EQU VAL2+1 ;VALUE 3RD FRACTION	
11896	003F	.	.	TMP4	EQU VAL3+1 ;TEMPORARY STORAGE	
11897	BD2F	.	.	.	; INIT SUBROUTINE ENTRY POINT	
11898	BD2F	2E	2F	.	INIT MVI L,PREX ;TO ADDR LAST WD TO MOVE	
11899	BD31	26	BD	.	INIT1 MVI H,ARTHB ;TO ADDRESS ROM COPY	
11900	BD33	5E	.	.	MOV E,M ;CURRENT WORD OF ROM COP	
11901	BD34	26	90	.	MVI H,SCRB ;TO ADDRESS RAM COPY	
11902	BD36	73	.	.	MOV M,E ;WRITE CURRENT WD TO RAM	
11903	BD37	2D	.	.	DCR L ;DECREMENT WORD ADDRESS	
11904	BD38	F2	31	BD	JP INIT1 ;IF MORE TO MOVE	
11905	BD3B	C9	.	.	RET ;RETURN TO CALLER	
11906	BD3C	.	.	.	; STR SUBROUTINE ENTRY POINT.	

				=====				PAGE 317
ITEM	LOC	OBJECT CODE	SOURCE	STATEMENTS				
=====								
11907	BD3C	73 . .	STR0	MOV M,E	;STORE ZEROETH WORD			
11908	BD3D	2C . .		INR L	;TO ADDRESS FIRST WORD			
11909	BD3E	77 . .	STR	MOV M,A	;STORE FIRST WORD			
11910	BD3F	2C . .	STR1	INR L	;TO ADDRESS SECOND WORD			
11911	BD40	70 . .		MOV M,B	;STORE SECOND WORD			
11912	BD41	2C . .		INR L	;TO ADDRESS THIRD WORD			
11913	BD42	71 . .		MOV M,C	;STORE THIRD WORD			
11914	BD43	2C . .		INR L	;TO ADDRESS FOURTH WORD			
11915	BD44	72 . .		MOV M,D	;STORE FOURTH WORD			
11916	BD45	C9 . .		RET	;RETURN TO CALLER			
11917	BD46	. . .	; FLOATING	POINT ZRO	SUBROUTINE ENT. PNT.			
11918	BD46	26 90 .	ZRO	MVI H,SCRB	;TO ADDRESS SCRATCH BANK			
11919	BD48	2E 30 .	ZRO1	MVI L,ACCE	;TO ADDR ACCUM EXPONENT			
11920	BD4A	AF . .		XRA A	;ZERO			
11921	BD4B	77 . .		MOV M,A	;CLEAR ACCUMULATOR EXPON			
11922	BD4C	C9 . .		RET	;RETURN TO CALLER			
11923	BD4D	. . .	; FLOATING	POINT CHS	SUBROUTINE ENT. PNT.			
11924	BD4D	3E 80 .	CHS	MVI A,200Q	;MASK FOR SIGN BIT			
11925	BD4F	0E . .		DB 016Q	;LBI INST TO SKIP NEXT WD			
11926	BD50	. . .	; FLOATING	POINT ABS	SUBROUTINE ENT. PNT.			
11927	BD50	AF . .	ABS	XRA A	;ZERO			
11928	BD51	26 90 .		MVI H,SCRB	;TO ADDRESS SCRATCH BANK			
11929	BD53	2E 31 .		MVI L,ACCS	;TO ADDRESS ACCUM SIGN			
11930	BD55	A6 . .		ANA M	;COMPLEMENT OF SIGN			
11931	BD56	EE 80 .		XRI 200Q	;COMPLEMENT THE SIGN BIT			
11932	BD58	77 . .		MOV M,A	;ACCUMULATOR SIGN			
11933	BD59	. . .	; FLOATING	POINT TEST	ENTRY POINT.			
11934	BD59	26 90 .	TST	MVI H,SCRB	;TO ADDRESS SCRATCH BANK			
11935	BD5B	2E 30 .	TST1	MVI L,ACCE	;TO ADDR ACCUM EXPONENT			
11936	BD5D	7E . .		MOV A,M	;ACCUMULATOR EXPONENT			
11937	BD5E	A7 . .		ANA A	;SET CONTROL BITS			
11938	BD5F	CA 46 BD		JZ ZRO	;IF ACCUMULATOR IS ZERO			
11939	BD62	5F . .		MOV E,A	;ACCUMULATOR EXPONENT			
11940	BD63	2C . .		INR L	;TO ADDR ACCUMULATOR SIGN			
11941	BD64	7E . .		MOV A,M	;ACCUMULATOR SIGN			
11942	BD65	2C . .		INR L	;TO ADDR ACCUM 1ST FRCTN			
11943	BD66	AE . .		XRA M	;ACCUM SIGN AND 1ST FRCTN			
11944	BD67	2C . .		INR L	;TO ADDR ACCUM 2ND FRCTN			
11945	BD68	4E . .		MOV C,M	;ACCUMULATOR 2ND FRACTIO			
11946	BD69	2C . .		INR L	;TO ADDR ACCUM 3RD FRCTN			
11947	BD6A	56 . .		MOV D,M	;ACCUMULATOR 3RD FRCTN			
11948	BD6B	C3 79 BE		JMP ADD12	;TO SET EXIT CONDITIONS			
11949	BD6E	. . .	; FLOATING	POINT LOAD	ENTRY POINT.			
11950	BD6E	7E . .	LOD	MOV A,M	;OPERAND EXPONENT			
11951	BD6F	A7 . .		ANA A	;SET CONTROL BITS			
11952	BD70	CA 46 BD		JZ ZRO	;IF OPERAND IS ZERO			
11953	BD73	5F . .		MOV E,A	;OPERAND EXPONENT			
11954	BD74	2C . .		INR L	;TO ADDR OP SIGN AND 1ST			
11955	BD75	7E . .		MOV A,M	;OPERAND SIGN AND 1ST FR			
11956	BD76	2C . .		INR L	;TO ADDRESS OPERAND 2ND FR			

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 318
11957	BD77	4E	.	MOV C,M ;OPERAND 2ND FRACTION	
11958	BD78	2C	.	INR L ;TO ADDRESS OPERAND 3RD FR	
11959	BD79	56	.	MOV D,M ;OPERAND 3RD FRACTION	
11960	BD7A	.	.	; STORE THE OPERAND IN THE ACCUMULATOR.	
11961	BD7A	6F	.	MOV L,A ;OPERAND SIGN AND 1ST FR	
11962	BD7B	F6	80	LOD1 ORI 200Q ;ACCUMULATOR 1ST FRACTION	
11963	BD7D	47	.	MOV B,A ;ACCUMULATOR 1ST FRACTIO	
11964	BD7E	AD	.	XRA L ;ACCUMULATOR SIGN	
11965	BD7F	26	90	MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
11966	BD81	2E	30	MVI L,ACCE ;TO ADDR ACCUM EXPONENT	
11967	BD83	CD	3C	BD CALL STRO ;SET THE ACCUMULATOR	
11968	BD86	A8	.	XRA B ;ACCUM SIGN AND 1ST FRCTN	
11969	BD87	.	.	; SET CONTROL BITS AND EXIT	
11970	BD87	47	.	MOV B,A ;ACCUM SIGN AND 1ST FRAC	
11971	BD88	F6	01	ORI 1 ;SET SIGN BIT FOR EXIT	
11972	BD8A	7B	.	MOV A,E ;ACCUMULATOR EXPONENT	
11973	BD8B	C9	.	RET ;RETURN TO CALLER	
11974	BD8C	.	.	; FLOATING POINT MUL SUBROUTINE ENT. PNT.	
11975	BD8C	7E	.	MUL MOV A,M ;OPERAND EXPONENT	
11976	BD8D	A7	.	ANA A ;SET CONTROL BITS	
11977	BD8E	C4	93	BE CNZ MDEX ;READ OPERAND IF NOT ZERO	
11978	BD91	CA	46	BD JZ ZRO ;IF ZERO OR UNDERFLOW	
11979	BD94	DA	CA	BD JC OVERF ;IF OVERFLOW	
11980	BD97	CD	4B	BF CALL MULX ;CALL FIXED MULT SUBRTN	
11981	BD9A	.	.	; NORMALIZE IF NECESSARY.	
11982	BD9A	78	.	MOV A,B ;1ST PRODUCT	
11983	BD9B	A7	.	ANA A ;SET CONTROL BITS	
11984	BD9C	FA	A9	BD JM RNDA ;IF NO NORMALIZATION REQUIRE	
11985	BD9F	2E	30	MVI L,ACCE ;TO ADDR ACCUM EXPONENT	
11986	BDA1	7E	.	MOV A,M ;ACCUMULATOR EXPONENT	
11987	BDA2	DE	01	SBI 1 ;DECREMENT ACCUMULATOR EXPON	
11988	BDA4	77	.	MOV M,A ;ACCUMULATOR EXPONENT	
11989	BDA5	C8	.	RZ ;RETURN TO CALLER IF UNDERF	
11990	BDA6	CD	BA	BE CALL LSH ;CALL LEFT SHIFT SUBROUTINE	
11991	BDA9	.	.	; ROUND IF NECESSARY.	
11992	BDA9	CD	2E	BF RNDA CALL ROND ;CALL ROUNDING SUBROUTINE	
11993	BDAC	DA	CA	BD JC OVERF ;IF OVERFLOW	
11994	BDAF	47	.	MOV B,A ;ACCUM SIGN AND 1ST FRAC	
11995	BDB0	F6	01	ORI 1 ;SET SIGN BIT	
11996	BDB2	7B	.	MOV A,E ;ACCUMULATOR EXPONENT	
11997	BDB3	C9	.	RET ;RETURN TO CALLER	
11998	BDB4	.	.	; FLOATING POINT DIV SUBROUTINE ENT. PNT.	
11999	BDB4	AF	.	DIV XRA A ;ZERO	
12000	BDB5	96	.	SUB M ;COMPLEMENT OF DIVISOR EXP	
12001	RDB6	FE	01	CPI 1 ;SET CARRY IF DIVISION BY ZE	
12002	BDB8	D4	93	BE CNC MDEX ;READ OPERAND IF NOT ZERO	
12003	BDBB	DA	CA	BD JC OVERF ;IF OVERFLOW OR DIVISION BY	
12004	RDBE	CA	4B	BD JZ ZR01 ;IF UNDERFLOW OR ZERO	
12005	BDC1	4F	.	MOV C,A ;DIVISOR 1ST FRACTION	
12006	BDC2	CD	8E	BF CALL DIVX ;CALL FIXED DIV SUBRTN	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 319
12007	BDC5	26 90 .	MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
12008	BDC7	DA A9 BD	JC RNDA ;IF NU OVERFLOW	
12009	BDCA	. . .	; SET OVERFLOW FLAG.	
12010	BDCA	26 90 .	OVERF MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
12011	BDC C	2E 2E .	MVI L,OVER ;TO ADDR OVERFLOW FLAG	
12012	BDCE	3E FF .	MVI A,377Q ;OVERFLOW FLAG	
12013	BDD0	77 . .	MOV M,A ;OVERFLOW FLAG	
12014	BDD1	07 . .	RLC ;SET CARRY BIT FOR EXIT	
12015	BDD2	C9 . .	RET ;RETURN TO CALLER	
12016	BDD3	. . .	; FLOATING POINT SUB SUBROUTINE ENT. PNT.	
12017	BDD3	3E 80 .	SB MVI A,200Q ;MASK TO CHANGE OP SIGN	
12018	BDD5	0E . .	DB '016Q ;LBI INST TO SKIP NEXT WD	
12019	BDD6	. . .	; FLOATING POINT ADD SUBROUTINE ENT. PNT.	
12020	BDD6	AF . .	AD XRA A ;ZERO	
12021	BDD7	. . .	; LOAD THE OPERAND.	
12022	BDD7	5E . .	MOV E,M ;OPERAND EXPONENT	
12023	BDD8	2C . .	INR L ;TO ADDR OP SIGN, 1ST FRCT	
12024	BDD9	AE . .	XRA M ;OPERAND SIGN AND 1ST FRCT	
12025	BDDA	47 . .	MOV B,A ;OPERAND SIGN AND 1ST FR	
12026	Bddb	2C . .	INR L ;TO ADDR OPERAND 2ND	
12027	BDDC	4E . .	MOV C,M ;OPERAND 2ND FRACTION	
12028	BDDD	2C . .	INR L ;TO ADDR OPERAND 3RD FRCTN	
12029	BDD E	56 . .	MOV D,M ;OPERAND 3RD FRACTION	
12030	BDDF	. . .	; SAVE INITIAL EXPONENT.	
12031	BDDF	26 90 .	MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
12032	BDE1	2E 30 .	MVI L,ACCE ;TO ADDR ACCUM EXPONENT	
12033	BDE3	7E . .	MOV A,M ;ACCUMULATOR EXPONENT	
12034	BDE4	2D . .	DCR L ;TO ADDR INITIAL EXPONENT	
12035	BDE5	77 . .	MOV M,A ;INITIAL EXPONENT	
12036	BDE6	. . .	; CHECK FOR ZERO OPERAND.	
12037	BDE6	7B . .	MOV A,E ;OPERAND EXPONENT	
12038	BDE7	A7 . .	ANA A ;SET CONTROL BITS	
12039	BDE8	CA 5B BD	JZ TST1 ;IF OPERAND IS ZERO	
12040	BDEB	. . .	; GENERATE SUBTRACTION FLAG, RESTORE	
12041	BDEB	. . .	; SUPPRESSED FRACTION BIT.	
12042	BDEB	68 . .	MOV L,B ;OPERAND SIGN AND 1ST FR	
12043	BDEC	78 . .	MOV A,B ;OPERAND SIGN AND 1ST FR	
12044	BDED	F6 80 .	ORI 200Q ;OPERAND 1ST FRACTION	
12045	BDEF	47 . .	MOV B,A ;OPERAND 1ST FRACTION	
12046	BDF0	AD . .	XRA L ;OPERAND SIGN	
12047	BDF1	2E 31 .	MVI L,ACCS ;TO ADDRESS ACCUMULATOR SI	
12048	BDF3	AE . .	XRA M ;SUBTRACTION FLAG	
12049	BDF4	2E 35 .	MVI L,SF ;TO ADDRESS SUBTRACTION FL	
12050	BDF6	77 . .	MOV M,A ;SUBTRACTION FLAG	
12051	BDF7	. . .	; DETERMINE RELATIVE MAGNITUDES OF	
12052	BDF7	. . .	; OPEKAND AND ACCUMULATOR.	
12053	BDF7	2E 30 .	MVI L,ACCE ;TO ADDRESS ACCUMULATOR EX	
12054	BDF9	7E . .	MOV A,M ;ACCUMULATOR EXPONENT	
12055	BDF A	A7 . .	ANA A ;SET CONTROL BITS	
12056	BDF8	CA 85 BE	JZ ADD17 ;IF ACCUMULATOR IS ZERO	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 320
12057	BDFE	93	.	.	SUB E ;DIFFERENCE IN EXPONENTS	
12058	BDFE	DA	0D	BE	JC ADD2 ;IF ACCUM SMALLER THAN OP	
12059	BE02	.	.	.	; CHECK FOR INSIGNIFICANT OPERAND.	
12060	BE02	FA	5B	BD	JM TST1 ;IF THE OPERAND IS INSIGNIFI	
12061	BE05	FE	19	.	CPI 031Q ;COMPARE SHIFT COUNT TO 25	
12062	BE07	DA	2C	BE	JC ADD3 ;JOIN EXCH PATH IF OP SIGNIF	
12063	BE0A	C3	5B	BD	JMP TST1 ;OPERAND IS INSIGNIFICANT	
12064	BE0D	.	.	.	; CHECK FOR INSIGNIFICANT ACCUMULATOR	
12065	BE0D	F2	85	BE	ADD2 JP ADD17 ;IF ACCUM IS INSIGNIFICANT	
12066	BE10	FE	E7	.	CPI 347Q ;COMPARE SHIFT COUNT TO MINU	
12067	BE12	DA	85	BE	JC ADD17 ;IF ACCUM IS INSIGNIFICANT	
12068	BE15	73	.	.	MOV M,E ;OPERAND EXPONENT	
12069	BE16	5F	.	.	MOV E,A ;SHIFT COUNT	
12070	BE17	2E	35	.	MVI L,SF ;TO ADDRESS THE SUBTRACTIO	
12071	BE19	7E	.	.	MOV A,M ;SUBTRACTION FLAG	
12072	BE1A	2E	31	.	MVI L,ACCS ;TO ADDRESS THE ACCUMULATO	
12073	BE1C	AE	.	.	XRA M ;OPERAND SIGN	
12074	BE1D	77	.	.	MOV M,A ;ACCUMULATOR SIGN	
12075	BE1E	AF	.	.	XRA A ;ZERO	
12076	BE1F	93	.	.	SUB E ;COMPLEMENT SHIFT COUNT	
12077	BE20	.	.	.	; EXCHANGE ACCUMULATOR AND OPERAND.	
12078	BE20	2C	.	.	INR L ;TO ADDR ACCUM 1ST FRACTIO	
12079	BE21	5E	.	.	MOV E,M ;ACCUMULATOR 1ST FRACTIO	
12080	BE22	70	.	.	MOV M,B ;OPERAND 1ST FRACTION	
12081	BE23	43	.	.	MOV B,E ;ACCUMULATOR 1ST FRACTIO	
12082	BE24	2C	.	.	INR L ;TO ADDR ACCUM 2ND FRACTIO	
12083	BE25	5E	.	.	MOV E,M ;ACCUMULATOR 2ND FRACTIO	
12084	BE26	71	.	.	MOV M,C ;OPERAND 2ND FRACTION	
12085	BE27	4B	.	.	MOV C,E ;ACCUMULATOR 2ND FRACTIO	
12086	BE28	2C	.	.	INR L ;TO ADDR ACCUM 3RD FRACTIO	
12087	BE29	5E	.	.	MOV E,M ;ACCUMULATOR 3RD FRACTIO	
12088	BE2A	72	.	.	MOV M,D ;OPERAND 3RD FRACTION	
12089	BE2B	53	.	.	MOV D,E ;ACCUMULATOR 3RD FRACTIO	
12090	BE2C	.	.	.	; POSITION THE OPERAND.	
12091	BE2C	CD	C7	BE	ADD3 CALL RSH ;POSITION THE OPERAND	
12092	BE2F	2E	35	.	MVI L,SF ;TO ADDRESS SUBTRACTION FL	
12093	BE31	7E	.	.	MOV A,M ;SUBTRACTION FLAG	
12094	BE32	A7	.	.	ANA A ;SET CONTROL BITS	
12095	BE33	2E	34	.	MVI L,ACC3 ;TO ADDR ACCUM 3RD FRCTN	
12096	BE35	FA	5C	BE	JM ADD9 ;IF SUBTRACTION REQUIRED	
12097	BE38	.	.	.	; ADD ADDEND TO AUGEND.	
12098	BE38	7E	.	.	MOV A,M ;AUGEND 3RD FRACTION	
12099	BE39	82	.	.	ADD D ;ADDEND 3RD FRACTION	
12100	BE3A	57	.	.	MOV D,A ;SUM 3RD FRACTION	
12101	BE3B	2D	.	.	DCR L ;TO ADDRESS AUGEND 2ND FRA	
12102	BE3C	7E	.	.	MOV A,M ;AUGEND 2ND FRACTION	
12103	BE3D	89	.	.	ADC C ;ADDEND 2ND FRACTION	
12104	BE3E	4F	.	.	MOV C,A ;SUM 2ND FRACTION	
12105	BE3F	2D	.	.	DCR L ;TO ADDRESS AUGEND 1ST FRA	
12106	BE40	7E	.	.	MOV A,M ;AUGEND 1ST FRACTION	

=====					PAGE 321	
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	
=====						
12107	BE41	88	.	.	ADC B	;ADDEND 1ST FRACTION
12108	BE42	47	.	.	MOV B,A	;SUM 1ST FRACTION
12109	BE43	D2	73	BE	JNC ADD11	;IF NO CARRY FROM 1ST FRCTN
12110	BE46	.	.	.	;	RIGHT SHIFT SUM TO NORMALIZED POSITION.
12111	BE46	1F	.	.	RAR	;RIGHT SHIFT SUM 1ST FRACTI
12112	BE47	47	.	.	MOV B,A	;SUM 1ST FRACTION
12113	BE48	79	.	.	MOV A,C	;SUM 2ND FRACTION
12114	BE49	1F	.	.	RAR	;RIGHT SHIFT SUM 2ND FRACTI
12115	BE4A	4F	.	.	MOV C,A	;SUM 2ND FRACTION
12116	BE4B	7A	.	.	MOV A,D	;SUM 3RD FRACTION
12117	BE4C	1F	.	.	RAR	;RIGHT SHIFT SUM 3RD FRACTI
12118	BE4D	57	.	.	MOV D,A	;SUM 3RD FRACTION
12119	BE4E	1F	.	.	RAR	;4TH FRCTN = LOW BIT OF 3RD
12120	BE4F	5F	.	.	MOV E,A	;SUM 4TH FRACTION
12121	BE50	2E	30	.	MVI L,ACCE	;TO ADDRESS ACCUMULATOR EX
12122	BE52	7E	.	.	MOV A,M	;ACCUMULATOR EXPONENT
12123	BE53	C6	01	.	ADI 1	;INCREMENT ACCUMULATOR EXPON
12124	BE55	DA	CA	BD	JC OVERF	;IF OVERFLOW
12125	BE58	77	.	.	MOV M,A	;ACCUMULATOR EXPONENT
12126	BE59	C3	73	BE	JMP ADD11	;TO ROUND FRACTION
12127	BE5C	.	.	.	;	SUBTRACT SUBTRAHEND FROM MINUEND.
12128	BE5C	AF	.	.	ADD9 XRA A	;MINUEND 4TH FRCTN IS ZERO
12129	BE5D	93	.	.	SUB E	;SUBTRAHEND 4TH FRACTION
12130	BE5E	5F	.	.	MOV E,A	;DIFFERENCE 4TH FRACTION
12131	BE5F	7E	.	.	MOV A,M	;MINUEND 3RD FRACTION
12132	BE60	9A	.	.	SBB D	;SUBTRAHEND 3RD FRACTION
12133	BE61	57	.	.	MOV D,A	;DIFFERENCE 3RD FRACTION
12134	BE62	2D	.	.	DCR L	;TO ADDRESS MINUEND 2ND FR
12135	BE63	7E	.	.	MOV A,M	;MINUEND 2ND FRACTION
12136	BE64	99	.	.	SBB C	;SUBTRAHEND 2ND FRACTION
12137	BE65	4F	.	.	MOV C,A	;DIFFERENCE 2ND FRACTION
12138	BE66	2D	.	.	DCR L	;TO ADDRESS MINUEND 1ST FR
12139	BE67	7E	.	.	MOV A,M	;MINUEND 1ST FRACTION
12140	BE68	98	.	.	SBB B	;SUBTRAHEND 1ST FRACTION
12141	BE69	47	.	.	MOV B,A	;DIFFERENCE 1ST FRACTION
12142	BE6A	DC	ED	BE	ADD10 CC	;COMPLEMENT IF NEGATIVE
12143	BE6D	F4	00	BF	CP	;NORMALIZE IF NECESSARY
12144	BE70	F2	48	BD	JP ZR01	;IF UNDERFLOW OR ZERO
12145	BE73	CD	2E	BF	ADD11 CALL	;CALL ROUNDING SUBROUTINE
12146	BE76	DA	CA	BD	JC OVERF	;IF OVERFLOW
12147	BE79	47	.	.	ADD12 MOV	B,A ;ACCUM SIGN AND 1ST FRCT
12148	BE7A	2E	2F	.	MVI L,PREX	;TO ADDRESS PREV EXPONENT
12149	BE7C	7B	.	.	MOV A,E	;ACCUMULATOR EXPONENT
12150	BE7D	96	.	.	SUB M	;DIFFERENCE IN EXPONENTS
12151	BE7E	6F	.	.	MOV L,A	;DIFFERENCE IN EXPONENTS
12152	BE7F	78	.	.	MOV A,B	;ACCUM SIGN AND 1ST FRCT
12153	BE80	F6	01	.	ORI 1	;SET SIGN BIT FOR EXIT
12154	BE82	7B	.	.	MOV A,E	;ACCUMULATOR EXPONENT
12155	BE83	5D	.	.	MOV E,L	;SIGNIFICANCE INDEX
12156	BE84	C9	.	.	RET	;RETURN TO CALLER

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
						322
12157	BE85	.	.	.	; LOAD THE ACCUMULATOR WITH THE OPERAND.	
12158	BE85	2E	35	.	ADD17 MVI L,SF ;TO ADDR SUBTRACTION FLAG	
12159	BE87	7E	.	.	MOV A,M ;SUBTRACTION FLAG	
12160	BE88	2E	31	.	MVI L,ACCS ;TO ADDR ACCUMULATOR SIGN	
12161	BE8A	AE	.	.	XRA M ;OPERAND SIGN	
12162	BE8B	2D	.	.	DCR L ;TO ADDR ACCUM EXPONENT	
12163	BE8C	CD	3C	BD	CALL STRO ;SET THE ACCUMULATOR	
12164	BE8F	A8	.	.	XRA B ;ACCUM SIGN AND 1ST FRCTN	
12165	BE90	C3	79	BE	JMP ADD12 ;JOIN EXIT CODE	
12166	BE93	.	.	.	; SUBROUTINE TO READ THE OPERAND AND	
12167	BE93	.	.	.	; CHECK THE ACCUMULATOR EXPONENT.	
12168	BE93	47	.	.	MDEX MOV B,A ;EXPONENT MODIFIER	
12169	BE94	2C	.	.	INR L ;TO ADDR OP SIGN, 1ST FRCT	
12170	BE95	4E	.	.	MOV C,M ;OPERAND SIGN AND 1ST FR	
12171	BE96	2C	.	.	INR L ;TO ADDRESS OPERAND 2ND FR	
12172	BE97	56	.	.	MOV D,M ;OPERAND 2ND FRACTION	
12173	BE98	2C	.	.	INR L ;TO ADDRESS OPERAND 3RD FR	
12174	BE99	5E	.	.	MOV E,M ;OPERAND 3RD FRACTION	
12175	BE9A	26	90	.	MVI H,SCRB ;TO ADDRESS SCRATCH BANK	
12176	BE9C	2E	30	.	MVI L,ACCE ;TO ADDRESS ACCUMULATOR EX	
12177	BE9E	7E	.	.	MOV A,M ;ACCUMULATOR EXPONENT	
12178	BE9F	A7	.	.	ANA A ;SET CONTROL BITS	
12179	BEA0	C8	.	.	RZ ;RETURN IF ACCUM IS ZERO	
12180	BEA1	80	.	.	ADD B ;RESULT EXPONENT PLUS BIAS	
12181	BEA2	47	.	.	MOV B,A ;RESULT EXPONENT PLUS BI	
12182	BEA3	1F	.	.	RAR ;CARRY TO SIGN	
12183	BEA4	A8	.	.	XRA B ;CARRY AND SIGN MUST DIFFE	
12184	BEA5	78	.	.	MOV A,B ;RESULT EXPONENT PLUS RI	
12185	BEA6	06	80	.	MVI B,200Q ;EXP BIAS, SIGN MASK, MS B	
12186	BEA8	F2	B6	BE	JP OVUN ;IF OVERFLOW OR UNDERFLOW	
12187	BEAB	90	.	.	SUB B ;REMOVE EXCESS EXP BIAS	
12188	BEAC	C8	.	.	RZ ;RETURN IF UNDERFLOW	
12189	BEAD	77	.	.	MOV M,A ;RESULT EXPONENT	
12190	BEAE	2C	.	.	INR L ;TO ADDRESS ACCUMULATOR SI	
12191	BEAF	7E	.	.	MOV A,M ;ACCUMULATOR SIGN	
12192	BEB0	A9	.	.	XRA C ;RESULT SIGN IN SIGN BIT	
12193	BEB1	A0	.	.	ANA B ;RESULT SIGN	
12194	BEB2	77	.	.	MOV M,A ;RESULT SIGN	
12195	BEB3	79	.	.	MOV A,C ;OPERAND SIGN AND 1ST FR	
12196	BEB4	B0	.	.	ORA B ;OPERAND 1ST FRACTION	
12197	BEB5	C9	.	.	RET ;RETURN TO CALLER	
12198	BEB6	07	.	.	OVUN RLC ;SET CARRY BIT IF OVERFLOW	
12199	BEB7	D8	.	.	RC ;RETURN IF OVERFLOW	
12200	BEB8	AF	.	.	XRA A ;ZERO	
12201	BEB9	C9	.	.	RET ;RETURN IF UNDERFLOW	
12202	BEBA	.	.	.	; SUBROUTINE TO LEFT SHIFT THE B, C,	
12203	BEBA	.	.	.	; D, AND E REGISTERS ONE BIT.	
12204	BEBA	7B	.	.	LSH MOV A,E ;ORIGINAL CONTENTS OF E	
12205	BEB8	17	.	.	RAL ;LEFT SHIFT E	
12206	BEBC	5F	.	.	MOV E,A ;RESTORE CONTENTS OF E R	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 323
12207	BEBD	7A . .	LSH1 MOV A,D ; ORIGINAL CONTENTS OF D	
12208	BEBE	17 . .	RAL ; LEFT SHIFT D	
12209	BEBF	57 . .	MOV D,A ; RESTORE CONTENTS OF D R	
12210	BEC0	79 . .	MOV A,C ; ORIGINAL CONTENTS OF C	
12211	BEC1	17 . .	RAL ; LEFT SHIFT C	
12212	BEC2	4F . .	MOV C,A ; RESTORE CONTENTS OF C R	
12213	BEC3	78 . .	MOV A,B ; ORIGINAL CONTENTS OF B	
12214	BEC4	8F . .	ADC A ; LEFT SHIFT B	
12215	BEC5	47 . .	MOV B,A ; RESTORE CONTENTS OF B R	
12216	BEC6	C9 . .	RET ; RETURN TO CALLER	
12217	BEC7	. . .	; RIGHT SHIFT THE B, C, D AND E REGISTERS	
12218	BEC7	. . .	; BY THE SHIFT COUNT IN THE A REGISTER	
12219	BEC7	. . .	; SHIFT OPERAND TO REGISTER INDICATED BY	
12220	BEC7	. . .	; SHIFT COUNT	
12221	BEC7	1E 00 .	RSH MVI E,0 ; OPERAND 4TH FRCTN IS ZERO	
12222	BEC9	2E 08 .	RSH0 MVI L,0100 ; EACH REG IS 8 BITS OF SHI	
12223	BECB	BD . .	RSH1 CMP L ; COMPARE SHIFT COUNT TO 8	
12224	BECB	FA D8 BE	JM RSH2 ; IF REQ SHIFT LESS THAN 8	
12225	BECF	5A . .	MOV E,D ; OPERAND 4TH FRACTION	
12226	BED0	51 . .	MOV D,C ; OPERAND 3RD FRACTION	
12227	BED1	48 . .	MOV C,B ; OPERAND 2ND FRACTION	
12228	BED2	06 00 .	MVI B,0 ; OPERAND 1ST FRACTION IS Z	
12229	BED4	95 . .	SUB L ; REDUCE SHIFT COUNT BY 1 R	
12230	BED5	C2 CB BE	JNZ RSH1 ; IF MORE SHIFTS REQUIRED	
12231	BED8	. . .	; SHIFT OPERAND RIGHT BY -SHIFT COUNT-	
12232	BED8	. . .	; BITS.	
12233	BED8	A7 . .	RSH2 ANA A ; SET CONTROL BITS	
12234	BED9	C8 . .	RZ ; RETURN IF SHIFT COMPLETE	
12235	BEDA	6F . .	MOV L,A ; SHIFT COUNT	
12236	BEDB	A7 . .	RSH3 ANA A ; CLEAR CARRY BIT	
12237	BEDC	78 . .	MOV A,B ; OPERAND 1ST FRACTION	
12238	BEDD	1F . .	RAR ; RIGHT SHIFT OP 1ST FRCTN	
12239	BEDE	47 . .	MOV B,A ; OPERAND 1ST FRACTION	
12240	BEDF	79 . .	MOV A,C ; OPERAND 2ND FRACTION	
12241	BEE0	1F . .	RAR ; RIGHT SHIFT OP 2ND FRCTN	
12242	BEE1	4F . .	MOV C,A ; OPERAND 2ND FRACTION	
12243	BEE2	7A . .	MOV A,D ; OPERAND 3RD FRACTION	
12244	BEE3	1F . .	RAR ; RIGHT SHIFT OP 3RD FRCTN	
12245	BEE4	57 . .	MOV D,A ; OPERAND 3RD FRACTION	
12246	BEE5	7B . .	MOV A,E ; OPERAND 4TH FRACTION	
12247	BEE6	1F . .	RAR ; RIGHT SHIFT OP 4TH FRCTN	
12248	BEE7	5F . .	MOV E,A ; OPERAND 4TH FRACTION	
12249	BEE8	2D . .	DCR L ; DECREMENT SHIFT COUNT	
12250	BEE9	C2 DB BE	JNZ RSH3 ; IF MORE SHIFTS REQUIRED	
12251	BEEC	C9 . .	RET ; RETURN TO CALLER	
12252	BEED	. . .	; COMPLEMENT THE B, C, D, AND E REGISTERS.	
12253	BEED	2D . .	COMP DCR L ; TO ADDR ACCUM SIGN	
12254	BEED	7E . .	MOV A,M ; ACCUMULATOR SIGN	
12255	BEEF	EE 80 .	XRI 2000 ; CHANGE SIGN	
12256	BEF1	77 . .	MOV M,A ; ACCUMULATOR SIGN	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 324
12257	BEF2	AF	.	COMP1	XRA A ;ZERO	
12258	BEF3	6F	.		MOV L,A ;ZERO	
12259	BEF4	93	.		SUB E ;COMPLEMENT 4TH FRCTN	
12260	BEF5	5F	.		MOV E,A ;4TH FRACTION	
12261	BEF6	7D	.		MOV A,L ;ZERO	
12262	BEF7	9A	.		SBB D ;COMPLEMENT 3RD FRCTN	
12263	BEF8	57	.		MOV D,A ;3RD FRACTION	
12264	BEF9	7D	.		MOV A,L ;ZERO	
12265	BEFA	99	.		SBB C ;COMPLEMENT 2ND FRCTN	
12266	BEFB	4F	.		MOV C,A ;2ND FRACTION	
12267	BEFC	7D	.		MOV A,L ;ZERO	
12268	BEFD	98	.		SBB B ;COMPLEMENT 1ST FRCTN	
12269	BEFE	47	.		MOV B,A ;1ST FRACTION	
12270	BEFF	C9	.		RET ;RETURN TO CALLER	
12271	BF00	.	.		; NORMALIZE THE REGISTERS.	
12272	BF00	2E	20	NORM	MVI L,0400 ;MAX NORMALIZING SHIFT	
12273	BF02	78	.	NORM1	MOV A,B ;1ST FRACTION	
12274	BF03	A7	.		ANA A ;SET CONTROL BITS	
12275	BF04	C2	20	BF	JNZ NORM3 ;IF 1ST FRACTION NONZERO	
12276	BF07	41	.		MOV B,C ;1ST FRACTION	
12277	BF08	4A	.		MOV C,D ;2ND FRACTION	
12278	BF09	53	.		MOV D,E ;3RD FRACTION	
12279	BF0A	5F	.		MOV E,A ;ZERO 4TH FRACTION	
12280	BF0B	7D	.		MOV A,L ;NORMALIZING SHIFT COUNT	
12281	BF0C	D6	08		SUI 0100 ;REDUCE SHIFT COUNT	
12282	BF0E	6F	.		MOV L,A ;NORMALIZING SHIFT COUNT	
12283	BF0F	C2	02	BF	JNZ NORM1 ;IF FRACTION NONZERO	
12284	BF12	C9	.		RET ;IF FRACTION IS ZERO	
12285	BF13	2D	.	NORM2	DCR L ;DECREMENT SHIFT COUNT	
12286	BF14	7B	.		MOV A,E ;ORIGINAL CONTENTS OF E	
12287	BF15	17	.		RAL ;LEFT SHIFT E	
12288	BF16	5F	.		MOV E,A ;RESTORE CONTENTS OF E R	
12289	BF17	7A	.		MOV A,D ;ORIGINAL CONTENTS OF D	
12290	BF18	17	.		RAL ;LEFT SHIFT D	
12291	BF19	57	.		MOV D,A ;RESTORE CONTENTS OF D R	
12292	BF1A	79	.		MOV A,C ;ORIGINAL CONTENTS OF C	
12293	BF1B	17	.		RAL ;LEFT SHIFT C	
12294	BF1C	4F	.		MOV C,A ;RESTORE CONTENTS OF C R	
12295	BF1D	78	.		MOV A,B ;ORIGINAL CONTENTS OF B	
12296	BF1E	8F	.		ADC A ;LEFT SHIFT B	
12297	BF1F	47	.		MOV B,A ;RESTORE CONTENTS OF B R	
12298	BF20	F2	13	BF	JP NORM2 ;IF NOT NORMALIZED	
12299	BF23	7D	.		MOV A,L ;NORMALIZING SHIFT COUNT	
12300	BF24	D6	20		SUI 0400 ;REMOVE BIAS	
12301	BF26	2E	30		MVI L,ACCE ;TO ADDR ACCUM EXPONENT	
12302	BF28	86	.		ADD M ;ADJUST ACCUM EXPONENT	
12303	BF29	77	.		MOV M,A ;NEW ACCUM EXPONENT	
12304	BF2A	C8	.		RZ ;RETURN IF ZERO EXP	
12305	BF2B	1F	.		RAR ;BORROW BIT TO SIGN	
12306	BF2C	A7	.		ANA A ;SET SIGN TO IND. U	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 325
12307	BF2D	C9	.	.	RET ;RETURN TO CALLER	
12308	BF2E	.	.	.	; SUBROUTINE TO ROUND THE B, C, D REGISTERS.	
12309	BF2E	2E	30	.	ROND MVI L,ACCE ;TO ADDR ACCUM EXPONENT	
12310	BF30	7B	.	.	MOV A,E ;4TH FRACTION	
12311	BF31	A7	.	.	ANA A ;SET CONTROL BITS	
12312	BF32	5E	.	.	MOV E,M ;ACCUMULATOR EXPONENT	
12313	BF33	FC	3D	BF	CM RNDR ;CALL 2ND LEVEL ROUNDER	
12314	BF36	D8	.	.	RC ;IF OVERFLOW	
12315	BF37	78	.	.	MOV A,B ;1ST FRACTION	
12316	BF38	2C	.	.	INR L ;TO ADDR ACCUM SIGN	
12317	BF39	AE	.	.	XRA M ;ACCUM SIGN AND 1ST FRCTN	
12318	BF3A	C3	3F	BD	JMP STR1 ;RETURN THRU STORE SUBR.	
12319	BF3D	.	.	.	; SECOND LEVEL ROUNDING SUBROUTINE.	
12320	BF3D	14	.	.	RNDR INR D ;ROUND 3RD FRACTION	
12321	BF3E	C0	.	.	RNZ ;RETURN IF NO CARRY	
12322	BF3F	0C	.	.	INR C ;CARRY TO 2ND FRACTION	
12323	BF40	C0	.	.	RNZ ;RETURN IF NO CARRY	
12324	BF41	04	.	.	INR B ;CARRY TO 1ST FRACTION	
12325	BF42	C0	.	.	RNZ ;RETURN IF NO CARRY	
12326	BF43	7B	.	.	MOV A,E ;ACCUMULATOR EXPONENT	
12327	BF44	C6	01	.	ADI 1 ;INCREMENT ACCUM EXPONENT	
12328	BF46	5F	.	.	MOV E,A ;NEW ACCUM EXPONENT	
12329	BF47	06	80	.	MVI B,200Q ;NEW 1ST FRACTION	
12330	BF49	77	.	.	MOV M,A ;NEW ACCUM EXPONENT	
12331	BF4A	C9	.	.	RET ;RETURN TO ROND SUBROUTINE	
12332	BF4B	.	.	.	; FIXED POINT MULTIPLY SUBROUTINE.	
12333	BF4B	2E	09	.	MULX MVI L,MULP1 ;TO ADDR 1ST MULTIPLICAND	
12334	BF4D	77	.	.	MOV M,A ;1ST MULTIPLICAND	
12335	BF4E	2E	05	.	MVI L,MULP2 ;TO ADDR 2ND MULTIPLICAND	
12336	BF50	72	.	.	MOV M,D ;2ND MULTIPLICAND	
12337	BF51	2E	01	.	MVI L,MULP3 ;TO ADDR 3RD MULTIPLICAND	
12338	BF53	73	.	.	MOV M,E ;3RD MULTIPLICAND	
12339	BF54	AF	.	.	XRA A ;CLEAR 6TH PRODUCT	
12340	BF55	5F	.	.	MOV E,A ;CLEAR 5TH PRODUCT	
12341	BF56	57	.	.	MOV D,A ;CLEAR 4TH PRODUCT	
12342	BF57	.	.	.	; MULTIPLY BY EACH ACCUMULATOR	
12343	BF57	.	.	.	; FRACTION IN TURN.	
12344	BF57	2E	34	.	MVI L,ACC3 ;TO ADDRESS 3RD FRCTN	
12345	BF59	CD	66	BF	CALL MULX2 ;MULTIPLY BY ACCUM 3RD FRCTN	
12346	BF5C	2E	33	.	MVI L,ACC2 ;TO ADDRESS 2ND FRCTN	
12347	BF5E	CD	63	BF	CALL MULX1 ;MULTIPLY BY ACCUM 2ND FRCTN	
12348	BF61	2E	32	.	MVI L,ACC1 ;TO ADDRESS 1ST FRCTN	
12349	BF63	.	.	.	; MULTIPLY BY ONE ACCUMULATOR WORD.	
12350	BF63	7A	.	.	MULX1 MOV A,D ;5TH PARTIAL PRODUCT	
12351	BF64	59	.	.	MOV E,C ;4TH PARTIAL PRODUCT	
12352	BF65	50	.	.	MOV D,B ;3RD PARTIAL PRODUCT	
12353	BF66	46	.	.	MULX2 MOV B,M ;MULTIPLIER	
12354	BF67	6F	.	.	MOV L,A ;5TH PARTIAL PRODUCT	
12355	BF68	AF	.	.	XRA A ;ZERO	
12356	BF69	4F	.	.	MOV C,A ;2ND PARTIAL PRODUCT	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 326
12357	BF6A	90 . .	SUB B ;SET CARRY BIT FOR EXIT FL	
12358	BF6B	DA 77 BF	JC MULX3 ;IF MULTIPLIER IS NOT ZERO	
12359	BF6E	4A . .	MOV C,D ;2ND PARTIAL PRODUCT	
12360	BF6F	53 . .	MOV D,E ;3RD PARTIAL PRODUCT	
12361	BF70	C9 . .	RET ;MULT BY ZERO COMPLETE	
12362	BF71	. . .	; COMPLETE ADDITION OF MULTIPLICAND.	
12363	BF71	4F . .	MULX5 MOV C,A ;2ND PARTIAL PRODUCT	
12364	BF72	D2 77 BF	JNC MULX3 ;IF NO CARRY TO 1ST PRODUCT	
12365	BF75	04 . .	INR B ;ADD CARRY TO 1ST PRODUCT	
12366	BF76	A7 . .	ANA A ;CLEAR CARRY BIT	
12367	BF77	. . .	; LOOP FOR EACH BIT OF MULTIPLIER WORD.	
12368	BF77	7D . .	MULX3 MOV A,L ;5TH PART PRODUCT, EXIT	
12369	BF78	8F . .	ADC A ;SHIFT EXIT FLAG OUT IF DO	
12370	BF79	C8 . .	RZ ;EXIT IF MULTIPLICATION DON	
12371	BF7A	6F . .	MOV L,A ;5TH PART PRODUCT, EXIT	
12372	BF7B	7B . .	MOV A,E ;4TH PARTIAL PRODUCT	
12373	BF7C	17 . .	RAL ;SHIFT 4TH PARTIAL PRODUCT	
12374	BF7D	5F . .	MOV E,A ;4TH PARTIAL PRODUCT	
12375	BF7E	7A . .	MOV A,D ;3RD PARTIAL PRODUCT	
12376	BF7F	17 . .	RAL ;SHIFT 3RD PARTIAL PRODUCT	
12377	BF80	57 . .	MOV D,A ;3RD PARTIAL PRODUCT	
12378	BF81	79 . .	MOV A,C ;2ND PARTIAL PRODUCT	
12379	BF82	17 . .	RAL ;SHIFT 2ND PARTIAL PRODUCT	
12380	BF83	4F . .	MOV C,A ;2ND PARTIAL PRODUCT	
12381	BF84	78 . .	MOV A,B ;1ST PART PROD AND MULTP	
12382	BF85	17 . .	RAL ;SHIFT 1ST PROD AND MULTIPL	
12383	BF86	47 . .	MOV B,A ;1ST PART PROD AND MULTI	
12384	BF87	D2 77 BF	JNC MULX3 ;IF NO ADDITION REQUIRED	
12385	BF8A	. . .	; ADD THE MULTIPLICAND TO THE PRODUCT	
12386	BF8A	. . .	; IF THE MULTIPLIER BIT IS ONE.	
12387	BF8A	7B . .	MOV A,E ;4TH PARTIAL PRODUCT	
12388	BF8B	C3 00 90	JMP MULX4 ;TO RAM CODE	
12389	BF8E	. . .	; FIXED POINT DIVIDE SUBROUTINE.	
12390	BF8E	. . .	; SUBTRACT DIVISOR FROM ACCUMULATOR TO	
12391	BF8E	. . .	; OBTAIN 1ST REMAINDER.	
12392	BF8E	2E 34 .	DIVX MVI L,ACC3 ;TO ADDRESS ACCUM 3RD FRCT	
12393	BF90	7E . .	MOV A,M ;ACCUMULATOR 3RD FRACTIO	
12394	BF91	93 . .	SUB E ;DIVISOR 3RD FRACTION	
12395	BF92	77 . .	MOV M,A ;REMAINDER 3RD FRACTION	
12396	BF93	2D . .	DCR L ;TO ADDRESS ACCUM 2ND FRCT	
12397	BF94	7E . .	MOV A,M ;ACCUMULATOR 2ND FRACTIO	
12398	BF95	9A . .	SBB D ;DIVISOR 2ND FRACTION	
12399	BF96	77 . .	MOV M,A ;REMAINDER 2ND FRACTION	
12400	BF97	2D . .	DCR L ;TO ADDRESS ACCUM 1ST FRCT	
12401	BF98	7E . .	MOV A,M ;ACCUMULATOR 1ST FRACTIO	
12402	BF99	99 . .	SBB C ;DIVISOR 1ST FRACTION	
12403	BF9A	77 . .	MOV M,A ;REMAINDER 1ST FRACTION	
12404	BF9B	. . .	; HALVE THE DIVISOR AND STORE FOR	
12405	BF9B	. . .	; ADDITION OR SUBTRACTION.	
12406	BF9B	79 . .	MOV A,C ;DIVISOR 1ST FRACTION	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 327
12407	BF9C	17 . .	RAL ;SET CARRY BIT	
12408	BF9D	79 . .	MOV A,C ;DIVISOR 1ST FRACTION	
12409	BF9E	1F . .	RAR ;HALF OF DIVISOR 1ST FRCTN	
12410	BF9F	. . . ;	+ 200B TO CORRECT QUOTIENT	
12411	BF9F	2E 19 .	MVI L,OP1S ;TO ADDRESS 1ST SUBTRACT D	
12412	BFA1	77 . .	MOV M,A ;1ST SUBTRACT DIVISOR	
12413	BFA2	2E 27 .	MVI L,OP1A ;TO ADDRESS 1ST ADD DIVISO	
12414	BFA4	77 . .	MOV M,A ;1ST ADD DIVISOR	
12415	BFA5	7A . .	MOV A,D ;DIVISOR 2ND FRACTION	
12416	BFA6	1F . .	RAR ;HALF OF DIVISOR 2ND FRACTI	
12417	BFA7	2E 15 .	MVI L,OP2S ;TO ADDRESS 2ND SUBTRACT D	
12418	BFA9	77 . .	MOV M,A ;2ND SUBTRACT DIVISOR	
12419	BFAA	2E 23 .	MVI L,OP2A ;TO ADDRESS 2ND ADD DIVISO	
12420	BFAC	77 . .	MOV M,A ;2ND ADD DIVISOR	
12421	BFAD	7B . .	MOV A,E ;DIVISOR 3RD FRACTION	
12422	BFAE	1F . .	RAR ;HALF OF DIVISOR 3RD FRACTI	
12423	BFAF	2E 11 .	MVI L,OP3S ;TO ADDRESS 3RD SUBTRACT D	
12424	BFB1	77 . .	MOV M,A ;3RD SUBTRACT DIVISOR	
12425	BFB2	2E 1F .	MVI L,OP3A ;TO ADDRESS 3RD ADD DIVISO	
12426	BFB4	77 . .	MOV M,A ;3RD ADD DIVISOR	
12427	BFB5	06 00 .	MVI B,0 ;INIT QUOTIENT 1ST FRCTN	
12428	BFB7	78 . .	MOV A,B ;DIVISOR FOURTH FRACTION	
12429	BFB8	1F . .	RAR ;LOW BIT OF DIVISOR 3RD FRA	
12430	BFB9	2E 0E .	MVI L,OP4S ;TO ADDRESS 4TH SUBTRACT D	
12431	BFB8	77 . .	MOV M,A ;4TH SUBTRACT DIVISOR	
12432	BFBC	2E 1C .	MVI L,OP4A ;TO ADDRESS 4TH ADD DIVISO	
12433	BFBE	77 . .	MOV M,A ;4TH ADD DIVISOR	
12434	BFBF	2E 2A .	MVI L,OP4X ;TO ADDRESS 4TH ADD DIVISO	
12435	BFC1	77 . .	MOV M,A ;4TH ADD DIVISOR	
12436	BFC2	. . . ;	LOAD 1ST REMAINDER, CHECK SIGN.	
12437	BFC2	2E 32 .	MVI L,ACC1 ;TO ADDR REMAINDER 1ST FRC	
12438	BFC4	7E . .	MOV A,M ;REMAINDER 1ST FRACTION	
12439	BFC5	2C . .	INR L ;TO ADDR REMAINDER 2ND FRC	
12440	BFC6	56 . .	MOV D,M ;REMAINDER 2ND FRACTION	
12441	BFC7	2C . .	INR L ;TO ADDR REMAINDER 3RD FRC	
12442	BFC8	5E . .	MOV E,M ;REMAINDER 3RD FRACTION	
12443	BFC9	A7 . .	ANA A ;SET CONTROL BITS	
12444	BFCA	FA F4 BF	JM DIVX4 ;IF REMAINDER IS NEGATIVE	
12445	BFC0	. . . ;	ADJUST EXPONENT, POSITION REMAINDER	
12446	BFC0	. . . ;	AND INITIALIZE THE QUOTIENT.	
12447	BFC0	2E 30 .	MVI L,ACCE ;TO ADDRESS ACCUMULATOR EX	
12448	BFCF	4E . .	MOV C,M ;QUOTIENT EXPONENT	
12449	BFD0	0C . .	INR C ;INCREMENT QUOTIENT EXPONE	
12450	BFD1	C8 . .	RZ ;RETURN IF OVERFLOW	
12451	BFD2	71 . .	MOV M,C ;QUOTIENT EXPONENT	
12452	BFD3	6B . .	MOV L,E ;REMAINDER 3RD FRACTION	
12453	BFD4	62 . .	MOV H,D ;REMAINDER 2ND FRACTION	
12454	BFD5	5F . .	MOV E,A ;REMAINDER 1ST FRACTION	
12455	BFD6	16 01 .	MVI D,1 ;INITIALIZE QUOT 3RD FRCTN	
12456	BFD8	48 . .	MOV C,B ;INITIALIZE QUOT 2ND FRC	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 328
12457	BFD9	. . .	; SUBTRACT THE DIVISOR FROM THE REMAINDER	
12458	BFD9	. . .	; IF IT IS POSITIVE	
12459	BFD9	AF . .	DIVX1 XRA A ;REMAINDER 4TH FRCTN IS ZE	
12460	BFDA	CD 0D 90	CALL DIVX5 ;CALL RAM SECTION	
12461	BFDD	07 . .	DIVX2 RLC ;SHFT REM 4TH FRCTN TO CY	
12462	BFDE	. . .	; SHIFT THE REMAINDER LEFT ONE BIT.	
12463	BFDE	78 . .	MOV A,B ;QUOTIENT 1ST FRACTION	
12464	BFDF	17 . .	RAL ;MS BIT OF QUOTIENT TO CY	
12465	BFE0	D8 . .	RC ;IF DIVISION COMPLETE	
12466	BFE1	1F . .	RAR ;REMAINDER 4TH FRCTN TO CY	
12467	BFE2	7D . .	MOV A,L ;REMAINDER 3RD FRACTION	
12468	BFE3	17 . .	RAL ;LEFT SHIFT REM 3RD FRCTN	
12469	BFE4	6F . .	MOV L,A ;REMAINDER 3RD FRACTION	
12470	BFES	7C . .	MOV A,H ;REMAINDER 2ND FRACTION	
12471	BFE6	17 . .	RAL ;LEFT SHIFT REM 2ND FRCTN	
12472	BFE7	67 . .	MOV H,A ;REMAINDER 2ND FRACTION	
12473	BFE8	CD BA BE	CALL LSH ;CALL LEFT SHIFT SUBROUTINE	
12474	BFEB	. . .	; BRANCH IF SUBTRACTION IS REQUIRED	
12475	BFEB	7A . .	MOV A,D ;QUOTIENT 3RD FRACTION	
12476	BFEC	0F . .	RRC ;REM SIGN INDIC TO CARRY BI	
12477	BFED	DA D9 BF	JC DIVX1 ;TO SUB DIVISOR IF REM POS	
12478	BFF0	. . .	; ADD THE DIVISOR IF THE REMAINDER	
12479	BFF0	. . .	; IS NEGATIVE.	
12480	BFF0	7D . .	DIVX3 MOV A,L ;REMAINDER 3RD FRACTION	
12481	BFF1	C3 1E 90	JMP DIVX6 ;TO RAM CODE	
12482	BFF4	. . .	; POSITION THE REMAINDER AND INITIALIZE	
12483	BFF4	. . .	; THE QUOTIENT.	
12484	BFF4	6B . .	DIVX4 MOV L,E ;REMAINDER 3RD FRACTION	
12485	BFF5	62 . .	MOV H,D ;REMAINDER 2ND FRACTION	
12486	BFF6	5F . .	MOV E,A ;REMAINDER 1ST FRACTION	
12487	BFF7	50 . .	MOV D,B ;INITIALIZE QUOT 3RD FRC	
12488	BFF8	48 . .	MOV C,B ;INITIALIZE QUOT 2ND FRC	
12489	BFF9	C3 F0 BF	JMP DIVX3 ;ADD DIVISOR IF REM IS NEG	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 329
=====
12491    BFFC      . . .      ;*****
12492    BFFC      . . .      ; END OF CODE
12493    BFFC      . . .      ;*****
12494    BFFC      . . .      END
0 ERRORS FOUND IN ASSEMBLY CODE
```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
ABFILL 7110 3708, 809
ABPARM 6487 1423, 1262
ABS BD50 11927, 8685, 8699, 9955, 9995, 10152, 10193, 10395, 11366, 11384,
11431, 11448, 11712
ABSFMT 63D1 1261, 743
ABSR0W B1DF 9829, 9563
ACBLOK 0081 348, 2557
ACC1 0032 11883, 11884, 11524, 12348, 12437
ACC2 0033 11884, 11885, 12346
ACC3 0034 11885, 11886, 12095, 12344, 12392
ACCE 0030 11881, 11882, 11501, 11513, 11553, 11919, 11935, 11966, 11985, 12032,
12053, 12121, 12176, 12301, 12309, 12447
ACCS 0031 11882, 11883, 11531, 11657, 11929, 12047, 12072, 12160
ACINHB 0040 184, 3546, 3554, 7289, 7677
ACOFF 7024 3553, 783
ACON 7013 3542, 782
ACON1 7019 3545, 3268, 3543
AD BDD6 12020, 10068, 10264, 11595, 11714
ADD10 BE6A 12142, 11509
ADD11 BE73 12145, 12109, 12126
ADD12 BE79 12147, 11948, 12165
ADD17 BE85 12158, 12056, 12065, 12067
ADD2 BE0D 12065, 12058
ADD3 BE2C 12091, 12062
ADD9 BE5C 12128, 12096
ADDCH1 AEA8 9123, 8926, 9146
ADDCHR AEA4 9120, 949, 950, 951, 952, 953, 954, 955, 956, 957,
958, 959, 960
ADDE AEC2 9144, 961, 962
ADDRTB AE49 9049, 9009
ADJCOL B923 11198, 9561, 9761
ADRH 0037 11888, 11889
ADRL 0036 11887, 11888, 11791, 11796
AF000 7187 3777, 3813
AF010 71C5 3811, 3805
AF015 71CB 3814, 3739
AF020 71CC 3816, 3751, 3795
AFILL1 713D 3730, 3854
AGC010 9E11 6305, 6276, 6278
AGCPOS 6F81 3454, 780
AJMPR 0001 148, 7743, 8372, 8426
ALTORG 9200 163, 164
ANC005 A2CB 7295, 7288, 7291
ANCHK A2B7 7282, 15, 3549
ANCTAB 9E14 6310, 6294
ANCUR 9DE5 6273, 50
ANG010 7674 4619, 4612
ANG013 76AB 4652, 4645
ANG015 76B2 4656, 4630, 4650
ANG020 7700 4703, 4675
ANG030 7739 4732, 4705, 4708, 4719, 4726
ANG040 774F 4750, 4672
ANG050 7759 4756, 4661, 4664, 4770

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
ANG055  7762  4766, 4761
ANG060  7764  4769, 4690, 4696
ANG065  7767  4771, 4718, 4725
ANGKEY  0096   317, 8066, 8131
ANGLE   7648  4586, 1101, 2256, 4561, 4855, 4873, 5514, 8011, 8108, 9877
ANGMSG  A72F   8110, 8031
ANGTAB  7770   4782, 4603
ANVOF1  6E29   3276, 3274, 5620, 7685, 7914
ANVOFF  6E23   3273, 771
ANVON   6E12   3262, 770
ANVON1  6E18   3265, 3263, 5534, 5928, 7913
APAXES  B1EF   9854, 7774, 8662
APB1    FBC3   518, 519, 648, 9051
APB10   FBA8   527, 528, 657, 9069
APB11   FBA4   528, 529, 658, 9071
APB12   FBA0   529, 530, 659, 9073
APB13   FB9E   530, 531, 660, 9075
APB14   FB9C   531, 532, 661, 9077
APB15   FB9A   532, 533, 662, 9079
APB16   FB98   533, 534, 663, 9081
APB2    FBC2   519, 520, 649, 9053
APB3    FBC1   520, 521, 650, 9055
APB4    FBC0   521, 522, 651, 9057
APB5    FBBC   522, 523, 652, 9059
APB6    FBB8   523, 524, 653, 9061
APB7    FBB4   524, 525, 654, 9063
APB8    FBB0   525, 526, 655, 9065
APB9    FBAC   526, 527, 656, 9067
APBUF   FAFE   560, 561, 9022, 9216, 9255
APC010  B913  11171,11154
APCHK   B8FA  11151, 25
APCR    AFBE   9377, 40
APDISP  0002   640,10941,11200
APE010  B9A2  11330,11407,11468
APE1    B9A6  11334,11337
APERR   B98D  11321, 9856,10882
APESC   AA35   8631, 1038, 1039
APEXIT  AA4D   8653, 1036, 1037, 1048, 8547, 8566, 8575, 8582, 8591, 8608,
      8614, 8616, 8624
APFLG2  FB97   534, 535, 8660, 8663, 9660, 9742,10055,10252,10474,10890,
      11063,11071,11127,11199
APFLGS  FB96   535, 539, 9312, 9378,10889,10957,11007,11045,11054,11137
APGSEQ  AD95   8899, 993, 994
APGTAB  6260   1010, 8900
APINIT  AA73   8675, 9863,10895
APIP    0002   629, 9382, 9591,10892,10959,11008,11138
APL005  B832  10951,10983
APL010  B836  10956,10993
APL020  B840  10962
APL030  B86B  10985,10974
APL040  B873  10990,10987
APL050  B87A  10995,10975
APL060  B87B  10997,10961

```

13255
 2648A MICROCODE LISTING 'GR70'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
APLABL 0002 244, 1157, 4994, 9888, 9908
APLAST FAE3 577, 582
APLF AFD0 9393, 52
APLTOF B882 11006, 1161, 1221, 4083, 5536, 7810, 8518, 8718, 9681,10946,
11173
APLTON B799 10867, 7773, 8665
APMENU AB3C 8804, 8877, 9083, 9084, 8769, 9050, 9052, 9054, 9056, 9058,
9060, 9062, 9064, 9066, 9068, 9070, 9072, 9074, 9076, 9078,
9080
APMSG1 B9B6 11343,11323,11401,11462
APMSG2 B9CA 11344,11325
APMSG3 B9D3 11345,11327
APMUOF AAD9 8727, 20, 7948, 9864,10885,11186
APMUON AAC9 8715, 7949
APNDX 629B 1043, 1033, 1035
APO010 B8AD 11036,11033
APOFST 0008 622, 8545
APS005 AF8D 9336
APS010 AFA7 9352, 9349
APSCAN AF6E 9311, 21, 9631
APSCN1 AF87 9329, 9321,10989
APSEQ A98F 8503, 696, 697, 1011, 1012
APTAB 6268 1017, 8504
APXERR B989 11312,11303,11369,11434
APYERR B98B 11313,11309,11387,11451
AREAGO 756A 4382, 4153
AREAPT 0004 180, 1712, 1882, 3759, 3804, 3910, 3947
ARITH BD00 11815,11816,11830,11832,11836,11840,11843,11845,11848,11852,
11856,11859,11861,11863,11867,11871,11874,11878
ARSTR 7570 4386, 4383
ARTHB 00BD 11816,11899
ASABFT 63B2 1233, 740
ASC010 63C1 1248, 1235, 1241
ASCABS 6455 1348, 1213, 1234
ASCEND 64A5 1401, 1252, 1283, 1299, 1304, 1309, 1317, 1332, 1335
ASCII 0040 246, 1215, 1255, 1286, 1403
ASCINC 6469 1361, 1240
ASCREL 6487 1381, 1247
ASINFT 63B8 1239, 741
ASRLFT 63BE 1246, 742
AVINHB 0020 183, 3266, 3282, 7289, 7682, 7912, 8261
B15 8000 299, 685, 688, 691, 694, 697, 700, 703, 706, 709,
712, 715, 722, 728, 730, 732, 753, 759, 761, 763,
792, 798, 800, 802, 833, 835, 837, 839, 848, 850,
852, 854, 856, 858, 860, 862, 866, 868, 870, 872,
874, 879, 881, 883, 888, 894, 896, 898, 905, 907,
909, 911, 913, 917, 919, 921, 929, 931, 933, 935,
937, 941, 943, 950, 952, 954, 956, 958, 960, 962,
966, 968, 970, 984, 986, 988, 990, 992, 994, 996,
998, 1000, 1012, 1014, 1019, 1021, 1023, 1025, 1027, 1029,
1031, 1037, 1039, 1041, 1056, 1058, 1060, 1062
BCDBIN AF4E 9268, 3038, 9243
BEGIN 6000 6, 2082, 8
  
```


13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'GR70'

SYMBOL VALUE REFERENCED ON

```

=====
BGNCUR FB94 539, 540, 9564, 9772
BGNPRM 6D06 3065, 3029, 3047, 3058
BNDCHK A341 7436, 2812, 7481, 7491
BNDCK1 A34A 7443, 6541, 6582, 6919, 6942, 7440, 7512, 7529
BNDCK2 A355 7460, 6909, 6930, 7502, 7521
BOTFLD 000F 611, 6391, 8602, 8916, 8946
BUSY 0001 189, 1911, 5897, 6613, 7228, 7234, 7244
CAPGO 7484 4276, 4148
CAPSTR 748A 4280, 4277
CCCHK 9DCC 6240, 6033, 6086, 6171, 6197, 6211, 6277
CFM1 9093 417, 418, 4606, 5345, 5347
CFXINC 9091 418, 419, 4614, 5382
CFYINC 908F 419, 420, 5386
CHAD BCEB 11795,11568,11611,11682,11741,11753,11774
CHEKAP B8F4 11136, 3569, 4526, 7029, 7921, 7937
CHF010 9884 5356, 5390
CHF020 98B4 5379, 5369
CHF030 98C5 5391, 5373
CHFIL1 9844 5326, 5705
CHFILL 983A 5319, 4959
CHKCH B919 11182, 969, 970
CHKCH1 B91D 11185, 9127
CHKMAX B95F 11281, 4998, 9964,10036,10161,10234
CHKMIN B955 11262, 9967,10041,10164,10239
CHKPTR 6AA9 2596, 2582, 2900
CHKTEK 6956 2344, 49, 2364, 2487, 2555, 2611, 2641, 2898, 7955
CHLEN FBD8 496, 497, 4626, 5343, 5700
CHPAT 9081 426, 427, 4957, 5362, 5374, 5376, 5702
CHRCNT FAEF 571, 572,10451,10641,10664,10681
CHROK 9802 5276, 4932
CHRTAB 7918 5070, 4955
CHS BD4D 11924
CKBPAT 9AD1 5732, 5701
CKSCLD 695C 2353, 2430, 2807, 2843, 4555, 4844, 4867, 4990, 5493, 7964,
7973, 8023, 8300
CKSLNT 7642 4575, 4313, 4611, 4629, 4684, 4713, 8025, 8049, 8099, 8327
CLAPF2 B8C2 11070, 8507, 8581, 8667, 8758, 9672, 9882,10644,11015
CLAPFL B8B5 11053, 9484, 9549, 9592
CLB010 A99F 8512, 8516
CLFLG1 A22C 7129, 1301, 1322, 3267, 3547, 3911, 7675, 7924
CLFLG3 A239 7146, 3501, 3536, 3584, 3618, 6469, 6510, 7027
CLFLG5 A246 7163, 1746, 3205, 3309, 3606, 5059, 7082
CLFLG6 A260 7197, 1740, 2252, 3001, 4550, 4569, 8094, 9874
CLFLG7 A26D 7215, 1158, 1200, 1287, 3069, 3108, 4370, 5578, 9909
CLIP 678E 2023, 1706
CLIPED 0080 185, 1730, 2025
CLM010 AA08 8599, 8605
CLM020 AA17 8606, 8603
CLOSE AEC7 9152, 8565, 8756, 8908, 8915, 8948, 8958
CLP010 67D6 2061, 2051
CLP020 6809 2091, 2065
CLPALG 683A 2126, 2057, 2078
CLR010 ADAF 8924, 8927

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```
=====
CLRBUF  A99A  8509, 8596, 8654
CLRERR  9BF5  5903, 5625, 5896
CLRFLD  ADAB  8921, 987, 988
CLRKEY  008D  347, 6130
CLRKY   A603  7954, 7767
CLRMEM  0009  295, 3170
CLRMNU  AA00  8595, 1047
CLRSMP  0013  335, 5627, 5629
CLRSUP  6BA9  2775, 32, 2380, 2766, 6241
CLRTAB  A0C6  6858, 6828
CLS010  AEDA  9160, 9157
CLS020  AEDD  9162, 9159
CLTKFL  A253  7180, 1154, 2237, 2371, 2419, 2490, 2647, 2691, 4235
CLX005  B119  9666, 9653
CLX010  B118  9670, 9662, 9744
CLX030  B133  9684, 9679
CLY010  B184  9748, 9735
CMMMA   002C  326, 4437
CNT1    906A  470, 471, 5324, 5371, 5703
CNT2    9069  471, 5328, 5367, 5378
CNTBF   FB9C  661,10923
CNTFLD  000D  607, 9293
CNTR    0008  232, 2253, 4929, 5414, 5458, 5508,10537,10572
COL1    FB03  557, 558, 8936, 8999, 9094, 9129
COLCNT  FB0C  547, 548, 9602, 9618,10888
COLTB   AE29  9029, 8996
COLX    B0EC  9644, 9607
COLY    B157  9726, 9611
COMMA   B068  9527, 9364
COMP    BEED  12253,11534,12142
COMP1   BEF2  12257
CONST   8907  270, 1900, 5641, 5857, 5895, 7616
CPA005  684D  2134, 2202, 2209, 2216
CPA010  6888  2175, 2167
CPA015  6897  2186, 2178
CPA020  68AB  2199, 2191
CPA030  68BA  2210, 2173, 2183, 2197
CPA040  68C9  2217, 2149
CPBLOK  0082  349, 2613
CPUPD1  982D  5306, 1146, 1670, 5432
CPUPDA  9832  5310, 2709, 4540, 4979, 5025, 5064, 5487, 5670, 5693
CR       000D  303
CRLF    A989  8493, 8377, 8431
CRLFON  75E4  4507, 1061, 1062
CRLFTB  62A6  1054, 4473
CURCD   90D2  382, 383, 1692, 1726, 2063, 3652, 3682
CURGCX  90CB  386, 387, 6491, 6511
CURGCV  90C9  387, 388, 6494, 6513
CURKEY  0088  323, 7843
CURMOD  90B5  391, 392, 1110, 1878, 3192, 3219, 3243, 3294, 3311, 3658,
        3662, 3677, 3774, 3876, 5351, 5745, 5776, 5778, 5795, 5826,
        5828, 6606, 7653, 7671, 7903, 8246,10373
CURPAT  90B4  392, 393, 1113, 1889, 3786, 3933, 3942, 3994, 7561, 7580
=====
```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
CURSAV  FB06    550, 551, 8744, 8761,10967,10972
CURTAB  A080    6795, 6764
CURTIC  FAF3     568, 569,10014,10065,10069,10212,10261,10265,10412,10427
D1      891E     255, 1807, 7622
D2      891C     256, 1820
DAP010  72FB     4011, 4018
DAP020  7309     4023, 4044
DAP030  730E     4026, 4041
DAP040  731A     4034, 4031
DAP050  731C     4036, 4033
DC      8912     261, 1824, 3765, 5344,10361
DCBYTE  FBE2     489, 490, 3373, 6998
DCNTRL  8904     273, 6999
DECPNT  0020     643,10475,10615,10638,10643
DEFAULT 997F     5530, 822
DEFAP   72EB     4003, 808
DEFLP   72AD     3970, 807
DEFLT1  9985     5533, 5531, 5730
DEFLT2  9988     5535, 5619
DELTAX  90D6     379, 380, 1551, 1773, 1776, 2105, 7554
DELTAY  90D4     380, 381, 1560, 1772, 2115, 7549
DELXY   658D     1571, 1550, 1559, 2104, 2114
DF1     A799     8147, 8145
DF11    A830     8251, 8245
DF12    A833     8253, 8249
DF13    A847     8265, 8259
DF14    A84A     8267, 8263
DF15    A853     8275, 8273
DF16    A85C     8283, 8281
DF17    A865     8292, 8290
DF18    A878     8304, 8298
DF18A   A87F     8306, 8302
DF19A   A8AA     8331, 8326
DF19B   A8AD     8333, 8329
DF2     A781     8159, 8153
DF20    A8F6     8379, 8361, 8404
DF3     A784     8161, 8157
DF4     A7C8     8173, 8167
DF5     A7CB     8175, 8171
DF6     A7DF     8188, 8182
DF7     A7E2     8190, 8186
DF9     A81C     8239, 8237
DFAPON  A856     8280, 7794
DFAVD   A836     8258, 7790
DFAXES  A85F     8289, 7795
DFCHK   9DD9     6252, 1179, 6244, 7715, 7740
DFCLR   A816     8236, 7788
DFDRAW  A90B     8401, 7791
DFGC    A7A0     8152, 7783
DFGVD   A81F     8244, 7789
DFM1    A8F9     8382, 7835, 8365, 8419
DFMENU  A84D     8272, 7793
DFMOV   A8C8     8357, 7792

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```
=====
```

DFRB	A7B7	8166, 7784
DFS010	A959	8445, 8455
DFS020	A95D	8451, 8462
DFSEND	A94D	8439, 8146, 8156, 8158, 8170, 8172, 8185, 8187, 8238, 8248, 8250, 8262, 8264, 8274, 8282, 8291, 8299, 8303, 8328, 8330
DFSND1	A950	8441, 8231, 8325, 8351, 8369, 8423
DFSTOP	A793	8144, 7782
DFSTX	A868	8297, 7796
DFTAB	A4F6	7781, 7749
DFTANG	A883	8312, 7797
DFTSIZ	A8B0	8338, 7798
DFZIN	A7E5	8195, 7786
DFZM	A7CE	8181, 7785
DFZMSZ	A802	8221, 8200, 8202, 8210
DFZOUT	A7F2	8206, 7787
DGC010	9F0E	6535, 6533
DGC020	9F4D	6575, 6573
DHL000	A320	7399, 7393
DHL010	A32E	7413, 7422
DHL020	A33B	7423, 7410, 7415
DIGIT	B033	9476, 9351
DINDX	60ED	765, 755, 757
DISPST	FFFE	141, 3277
DIV	BDB4	11999, 8691, 8705, 10005, 10203, 10401, 11365, 11383, 11430, 11447, 11667, 11698
DIV128	6C44	2917, 2453, 2926
DIVHL	A31C	7396, 6960, 6968
DIVHL1	A31A	7394, 2433, 4738, 5459
DIVHLR	A311	7389, 2137, 2141
DIVX	BF8E	12392, 12006
DIVX1	BFD9	12459, 12477
DIVX2	BFD0	12461, 11875
DIVX3	BFF0	12480, 12489
DIVX4	BFF4	12484, 12444
DIVX5	900D	11843, 12460
DIVX6	901E	11861, 12481
DOLLAR	B05E	9512, 9366
DRAWGC	9EFA	6521, 6495, 6514
DRAWRB	709B	3632, 3604
DRW010	6678	1770, 1758
DRW015	668F	1796, 1786
DRW020	66EF	1876, 1867
DRW030	670E	1893, 1888
DRWDOT	8901	271, 1906, 7619
DRWGC	0010	193, 6613, 7228, 7234
DRWVEC	6660	1754, 1716, 5849
DSPFLD	FAFC	561, 562, 6414, 8552, 9015, 9105, 9132, 9154
DSPFNC	0001	146, 6254
DSPGO	7454	4266, 4147
DSPMNU	AA1D	8612, 1049
DSPSEQ	6D6C	3159, 687, 688
DSPSTR	FE4F	140
DSPTAB	60D2	751, 3160

13255

2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
DSPTST 6369 1175, 37
DSPY 745A 4270, 4267
DWAXES A9F8 8588, 1046
DWFRST 0008 211, 1323, 1686, 1886, 1907, 3665, 3908, 3996, 5396, 5847
DYNGO 7576 4391, 4154
DYNSTR 757C 4395, 4392
ECNT FAEF 572, 573, 10449, 10505, 10530, 10548, 10559
ENAB0 9EAB 6458, 1163, 1341, 2381, 2491, 2517, 5583, 7673
ENABGC 9EB5 6468, 3195, 3232, 3821
EOD 0001 337, 6370, 6396
EOFRM 6F7B 3446, 3300, 3317, 3340, 5722, 5723, 6001
EOL 00CC 118, 5975, 5976, 6422, 8042, 8148, 8160, 8162, 8174, 8176,
8189, 8191, 8240, 8252, 8254, 8266, 8268, 8276, 8284, 8293,
8305, 8307, 8332, 8334, 8806, 8806, 8807, 8811, 8815, 8819,
8823, 8827, 8831, 8835, 8839, 8839, 8840, 8844, 8848, 8852,
8856, 8856, 8857, 8862, 8867, 8871, 8875

ESC 001B 338, 8148, 8305, 8464
ESCCH 6B81 2745, 887, 888
ESCSTR A96E 8463, 8442
EXC010 B76A 10828, 10832
EXC030 B772 10835, 10839, 10843
EXC040 B787 10848, 10854
EXC050 B791 10857, 10861
EXP B04A 9496, 9362
EXPCVT B744 10797, 10669
FAILCK 9BFD 5913, 5631, 5638
FILBF1 9E66 6402, 6353
FILBUF 9E50 6388, 6375
FIX BB17 11512, 11084, 11237, 11740
FIX1 BB45 11542, 11516, 11522
FLB010 9E59 6394
FLB020 9E63 6400, 6392
FLB030 9E86 6424, 6399
FLD1 0000 610, 1124, 8909, 8956
FLDTB AE39 9037, 8990
FLT BB00 11493, 11254, 11593, 11745
FMT010 63E6 1281, 1263, 1269, 1275
FORMAT A2D1 7311, 1436, 1443, 2422, 2425, 2625, 2628, 7339
FP1001 BA4A 11394, 11367, 11385
FP101 BACC 11454, 11432, 11449
FP359 AAC5 8710, 8702
FP719 AAC1 8709, 8688
FPCHK AEE8 9182, 9156
FPCNVT AF03 9205, 9161
FPINP B13F 9704, 9213, 9697
FPONE B14F 9712, 9706
FPSAV2 FAE4 575, 576, 10396, 10402
FPSAVE FAE8 574, 575, 10393, 10407
FPTWO B58A 10415, 10400
FRAME B286 9919, 9870
FRMFLD 000F 609, 9297
FROMBF FB98 663, 10936
FSTKEY 00A5 353, 6744, 6752, 6818, 6821

```

13255

2648A MICROCODE LISTING 'GR70'

SYMBOL VALUE REFERENCED ON

13255/90010

REV 04/17/78

```
=====
FSTRM2  9000      366, 369, 372, 1085
FSTVEC  FF06      316, 3255
FTEN    BCF6    11804,11583,11665,11694,11737
GC1DC   891E     280, 6552
GC1HI   8918     282, 6562
GC1LO   891A     281, 6561
GC2DC   891C     283, 6591
GC2HI   8914     285, 6603
GC2LO   8916     284, 6602
GCBLOK  0084     351, 2715
GCCHK   A981    8485, 7919, 7935, 8359, 8402
GCGO    73FE    4194, 4145
GCK010  A071    6782, 6771, 6780, 6794
GCK020  A078    6791, 6753
GCKEY   00A1     352, 6741, 6816
GCKEYS  A035    6739, 16
GCLEAR  6D72    3166, 766
GCLR1   6D78    3169, 1080, 2494, 3167, 5626, 5642, 5687, 5731, 7957
GCM005  A024    6717, 6714
GCM010  A02D    6723, 6721
GCM1    0001     208, 2660, 3212, 3468, 6689, 6892, 7593
GCM3    0020     213, 2660, 3468, 3573, 6689, 7080, 7593
GCM4    0040     214, 1745, 3468, 5055, 6689
GCMON   A00A    6696, 7050
GCMSG   A55F    7848, 7831
GCON    0040     203, 6480, 6488, 6504, 6509, 7058
GCORG   734E    4073, 816
GCP1    6F8F    3460, 3440, 3458
GCP2    6FA1    3467, 3490
GCTAB   9F8C    6622, 6651
GCTIMR  FBED     482, 483, 6700, 6773
GCWBG   740B    4209, 4100
GCWBLK  0004     332, 4099, 4221
GCWCH1  7417    4220, 4243
GCWCHR  7411    4217, 943
GCWESC  741C    4229, 940, 941, 942
GCWGO   743E    4253, 4146
GCWTAB  61F0     939, 4210, 4247
GCX     9061     451, 452, 6524, 6598
GCXY    9FCC    6644, 6730, 6776
GCY     905F     452, 453, 6522, 6556, 6566
GETFP   B135    9691, 9648, 9730
GETKEY  A428    7629, 6128, 7839, 7995, 8062, 8125,11335
GETLBL  B58E   10423,10097,10292
GETPAT  A3AB    7538, 1713
GETPRM  6D43    3118, 2276, 2288, 3335, 3866, 3901, 4095, 4847, 4870, 5496
GETPTR  FBC5     507, 508, 6372, 6377, 6405
GETVAL  A971    8470, 8387, 8392, 8410, 8415
GETVEC  64AF    1412, 1349, 1362, 1382
GETWA   676F    1973, 1767, 3782, 5359, 6560, 6600, 6988,10354
GEXIT   99C1    5571, 1257, 1288, 1302, 1307, 1315, 1319, 2277, 2279, 2290,
        2297, 2303, 3168, 3177, 3202, 3239, 3264, 3275, 3290, 3306,
        3336, 3338, 3341, 3436, 3457, 3459, 3477, 3491, 3498, 3523,
=====
```

13255
 2648A MICROCODE LISTING 'GR70'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
          3544, 3558, 3567, 3581, 3711, 3822, 3831, 3868, 3902, 3905,
          3973, 3976, 3979, 3981, 3991, 3998, 4006, 4045, 4053, 4061,
          4093, 4105, 4523, 4546, 4556, 4562, 4845, 4848, 4851, 4868,
          4871, 4874, 5494, 5497, 5500, 5532, 8622, 8657, 8668
GFLGS1  90B2  394, 395, 1103, 1677, 1711, 1728, 1756, 1862, 2024, 3654,
          3657, 3680, 3758, 3803, 3989, 4183, 7120, 7130, 7285, 7911,
          8260,11152
GFLGS2  90B1  395, 396, 1910, 3225, 3323, 3325, 6611, 7009, 7226
GFLGS3  90B0  396, 397, 3590, 3613, 4533, 4982, 5046, 6445, 6478, 6502,
          7053, 7068, 7084, 7139, 7147, 7821, 7855, 8154, 8168, 8486
GFLGS4  90AF  397, 398, 6648, 6697, 6759, 6767, 6830, 6832
GFLGS5  90AE  398, 399, 1104, 1885, 1904, 3663, 3666, 3675, 4341, 5054,
          6440, 6712, 6869, 6882, 6897, 7079, 7156, 7164, 7866, 7875,
          7891, 8183
GFLGS6  9097  415, 416, 1738, 2652, 3512, 3667, 3670, 3673, 4576, 4669,
          4928, 4985, 5050, 5413, 5457, 5507, 5510, 5658, 6229, 7190,
          7198,10536,10570,10573,10577
GFLGS7  9096  416, 417, 1402, 3027, 3044, 3056, 3079, 4366, 4993, 5550,
          5555, 7207, 7216
GGFLGS  FBC8  505, 506, 6346, 6369, 6395
GGINIT  9E24  6344, 45
GGT010  9E4C  6379, 6371
GGTEST  9E21  6334, 44
GINBLK  0083  350, 2675
GINCH   6B14  2670, 882, 883
GINCH1  6B1A  2673, 2734, 2747, 2764
GINCH2  6B1C  2676, 4222
GINCH3  6B37  2688, 2767
GINCHR  FAE2  582, 583, 2687, 2699, 4259
GINCR   6B9A  2762, 878, 879
GINENQ  6B5B  2714, 2754
GINESC  6B69  2726, 880, 881
GINMOD  0010  223, 1153, 2245, 2264, 2370, 2489, 2648, 2690, 3508, 4234,
          7701
GINTAB  61A0  877, 2643, 2756
GLAST   FBC4  509, 518
GLB010  B58A  10459,10457,10470
GLB020  B5C9  10471,10465
GLB030  B5D6  10481,10476
GLB040  B5D7  10483,10495,10499
GLB050  B5EF  10500,10492
GLB060  B5F7  10507,10515
GLB070  B602  10516,10489
GLB080  B60E  10527,10523
GLB090  B620  10539,10575
GLB100  B63B  10555,10538
GNE010  6B7B  2738, 2728
GNECTB  61AC  886, 2739
GOAP    A9E8  8572, 1044
GPARAM  6CA6  3008, 752, 753, 791, 792, 832, 833, 928, 929, 1413
GPM010  6CB3  3014, 3011
GPM020  6CC5  3025, 3013
GPM030  6CE3  3043, 3016, 3018
=====

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
GPM040 6CF9 3055, 3020
GPR010 6DSF 3138, 3132
GPT010 9AAF 5714, 5726
GRAM 90E0 372, 375
GRCOF1 9EE8 6506, 7087
GRCOFF 9EDE 6500, 3532, 6451
GRCON 9EB8 6476, 3516
GRCON1 9EC2 6482, 7071
GRDPAT 0088 620,10108,10304
GRESET 8960 252, 1072
GRGET 9E37 6367, 46
GRIDBF FB9A 662,10104,10299
GRIDFD 000E 608, 9299
GSBLOK 906B 438, 445, 2559, 2677, 2716, 4102, 4116
GSEND 697E 2379, 855, 856
GSET 6D7D 3175, 767
GSET1 6D83 3178, 3176, 5633
GSETUP 637A 1196, 14
GSMODE 0008 222, 2245, 2264, 2372, 2489, 2646
GST010 7378 4101, 4108
GST020 7384 4106, 4096, 4098
GSTAT 7361 4091, 930, 931
GTAB 6076 683, 1201, 8641
GTEST 99E2 5616, 48
GTEXT 0002 230, 2653, 3513, 4528, 4549, 6230
GTKST 6A98 2580, 2568, 2620
GTP010 A3D3 7563, 7558
GTP020 A3DA 7574, 7578
GTP030 A3E2 7579, 7576
GTPAT1 A3D6 7568, 7553, 7607
GTPAT2 A3FD 7599, 3760, 3810
GTS010 6AA2 2585, 2583
GTV010 A97D 8478, 8473
GTXMAX 0002 342, 1624, 2200
GTXOF 7618 4544, 785
GTXOF1 761E 4547, 4545, 7811, 9861,10883
GTXON 75F1 4521, 784
GTXON1 75F7 4524, 2247, 4522, 7962
GTYMAX 0008 344, 1657, 2176
GV005 6DC4 3216, 3210
GV010 6DDF 3229, 3227
GVENAB 0010 298, 3218, 3242, 3295, 3312, 3660, 7654, 7904, 8247
GVOFF 6DE7 3237, 769
GVOFF1 6DED 3240, 3238, 5927, 7905
GVON 6DA6 3200, 768
GVON1 6DAC 3203, 3201, 7906, 9865,10886
GWA010 6778 1981, 1979
GWE010 742D 4238, 4231
GXY010 9FE1 6664, 6667
GXY020 9FF0 6673, 6676
HAPAT FBF7 479, 480, 4009, 4025, 7552, 7606
HARD1 62D4 1081, 1075, 5553
HAVED 0010 632, 3030, 3068, 3081, 3107, 5577, 9485, 9516, 9532, 9594
=====

```


13255

2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
HAVEE      0004      630, 9435, 9464, 9500, 9503, 9516, 9536, 9548
HAVEP      0008      631, 3032, 3060, 3068, 3107, 5577, 9459, 9464, 9467, 9516,
                9536, 9548
HAVES      0040      634, 9411, 9435
HAVEX      0004      641, 9664, 9671, 9743
HAVEY      0008      642, 9661, 9671, 9746
HCEJK      8941      288, 1109, 1879, 3186, 3191, 3193, 3775, 3888, 5353, 6610,
                7648,10375
HEIGHT     9063      450, 451, 3740, 3791, 3796
HIL005     B19B      9771, 9768
HIL010     B1B4      9787, 9815
HIL015     B1C8      9807, 9799
HIL020     B1D6      9816, 9792, 9806, 9822
HIL030     B1D8      9820, 9776, 9778
HILITE     B186      9758, 9646, 9728
HISPD      0040      354, 6778, 6793
HIXY       6987      2386, 847, 848
HLINE      B506      10336, 9923, 9926, 9973,10278,10280,10296,10307,10313,10325
HOME       AD9B      8907, 47, 983, 984, 985, 986, 8607
HOMEDN     ADA3      8914, 989, 990
HRD010     62D9      1086, 1089
HRD015     62E4      1093, 1099
HRD020     6315      1117, 1121
HRD1       632B      1131, 1162,11013
HRD2       6340      1141,10908
HRDRST     62B9      1070, 12
HT010      9B5C      5805, 5812
HT020      9B6F      5816, 5807
HT030      9B90      5829, 5836
HTEST      9B41      5794, 5630, 5637
HWFLGS     8920      287, 1073, 1912, 3326, 5898, 5906, 7233
HWGO       6722      1909, 3789, 5365,10379
HWSTAT     8920      251, 5852, 5873, 7046, 7243, 7260
HXY010     6998      2396, 2389
ICN010     AF1B      9229, 9245, 9251
ICN020     AF3E      9246, 9233, 9235, 9237
ICN030     AF43      9252, 9239, 9241
ICNVRT     AF15      9225, 9158
IDBLOK     0001      331, 4107
IDGO       73D2      4166, 4143
IDSTR      73D8      4170, 4167
IGCPOS     6FA6      3474, 781
IGNCNT     FB05      551, 552, 9479,10891,11166
IGNSEQ     99DC      5594, 705, 706, 708, 709, 711, 712
IGNTAB     6180      892, 5595
ILEN       FBD1      500, 501
IMGX       FBCF      501, 502
IMGY       FBCD      502, 503
INC010     64EC      1460, 1457
INCFMT     63DD      1273, 745
INCPRM     6509      1485, 1274
INIT       BD2F      11898, 1123
INIT1      BD31      11899,11904

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

=====

INITD	8910	262, 1817
INP	BB4A	11549, 9705
INP1	BB69	11568,11560,11562,11599,11604,11608
INP10	BC0D	11670,11666
INP2	BB71	11574,11564
INP3	BRAE	11605,11576
INP4	BBB8	11611,11578
INP5	BBCB	11623,11619
INP6	BBCF	11629,11622
INP7	BBE8	11649,11639
INP8	BBEF	11654,11580,11609,11631,11651
INP9	BBF8	11661,11669,11673
INSERT	FB04	552, 557,10955,11202,11220,11222
INSFIX	B933	11218, 24
INT	B8C9	11082, 9654, 9736, 9959,10030,10156,10228
INT010	B8D8	11094,11092
INTCHK	AF5F	9291, 9258
INTDLY	0010	356, 6774
INTFLG	FFF6	153, 7257, 7263, 7266
IOFSTX	FBC8	503, 504
IOFSTY	FBC9	504, 505
IOKBCO	8380	157, 7098, 7105
IVOFF	0080	121, 8861, 8866
IVON	0082	120, 7849, 8017, 8111, 8113, 8135, 8805, 8809, 8813, 8817, 8821, 8825, 8829, 8833, 8837, 8842, 8846, 8850, 8854, 8859, 8864, 8869, 8873,11343
KBD010	A5DB	7925, 7941
KBDRAW	A5CE	7918, 7770, 8432
KBF000	A47F	7703, 7699
KBF005	A48B	7711, 7702, 7718
KBF010	A490	7714, 7707, 7710
KBF015	A49C	7719, 7716
KBF020	A4AE	7730, 7723, 7725
KBF030	A4C5	7747, 7741
KBF040	A4CB	7750, 7744
KBFUNC	A471	7695, 26
KBJMP1	FFFB	147, 7742, 8371, 8425
KBJMP2	FFFA	142, 2322
KBMOVE	A5EA	7934, 7771, 8378
KYBDTB	A4D4	7760, 7735
LABEL	0080	236, 2958, 3000, 4549, 4986, 5051, 6230
LABLX	B397	10093,10058
LABLY	B4C3	10289,10255
LAST	FADB	588, 592
LBF010	B673	10601,10612
LBF025	B682	10609,10597,10599
LBF030	B687	10613,10603
LBF040	B689	10617,10625
LBF050	B699	10626,10605,10607,10621,10623
LBF060	B6A9	10639,10631,10635
LBL1	6C7C	2966, 2974
LBL2	6C8B	2981, 2986
LBLBUF	FB0D	545, 5040, 5472, 5936, 5945, 5962, 5994

13255

2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
LBLCHK  BA76  11418, 9859
LBLENT  FAF2   569, 570
LBLECR  6C79  2964, 906, 907
LBLECR2 6C88  2979, 916, 917
LBLECTR 9074  433, 434, 2957, 2999, 5030, 5469, 5482, 5512, 6008, 7976,
      8028
LBLEN   0004  616,10100,10115,10117,10295,10310,10312
LBLEND  75E1  4500, 920, 921
LBLERR  BADO  11460, 9860
LBLESC  6C97  2991, 910, 911
LBLEFMT 8662  10591, 9898, 9906
LBLLF   6C82  2972, 908, 909
LBLLF2  6C91  2984, 918, 919
LBLOFF  6C9D  2997, 1156, 2982, 2992, 4501, 4530, 6352, 7748, 7830
LBLPTR  FAF0   570, 571,10454,10660,10670,10733
LBLERND B6B5  10657,10437
LBLESEQ 6C6A  2953, 699, 700
LBLETAB 61BC   903, 2954
LBLETB2 61D0   915, 2967
LBMSG1  BAE8  11470,11466
LBR010  B6C7  10672,10676
LBR020  B6D3  10684,10691
LBR030  B6DF  10696,10689
LBR040  B6E1  10698,10718
LBR050  B6FD  10716,10701,10714
LBR060  B701  10719,10704
LBR070  B70A  10732,10727
LBR080  B738  10775,10759
LBR090  B73F  10781,10778
LBX005  B3C6  10116,10107
LBX010  B3CC  10120,10085
LBY005  B4F1  10311,10303
LBY010  B4F7  10315,10281
LCK010  BAA1  11436,11424
LCT010  A559  7845, 7840
LEDCHK  9DDF  6261, 1183, 7644
LEN     0014  358, 6538, 6578
LFD010  9D62  6150, 6147
LFD020  9D72  6159, 6156
LFUPDA  9D55  6142, 6097, 6219
LIN010  B531  10367,10365
LINE1   B50C  10345,10338
LINEBF  FBC0   651,10930
LINEPT  0002  179, 3910, 3939, 3990
LINETB  729B  3953, 3930
LNF010  A2F5  7349, 7347
LNGFMT  A2E3  7336, 1497, 1503, 1522, 1530
LNTYPE  FADB   587, 588, 1667, 3926
LOCATE  A532  7829, 7820, 7847
LOD     BD6E  11950, 8682, 8689, 8696, 8703, 9710, 9946, 9951, 9991,10003,
      10066,10143,10148,10189,10201,10262,10399,10408,10428,11300,
      11306,11357,11361,11375,11379,11422,11426,11439,11443,11778
LOD1    BD7B  11962

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

=====

LORG	9944	5492, 821
LORG1	9959	5501, 5499, 5663, 5696, 9896, 9904, 9913
LORGTB	9976	5516, 5505
LOSPD	0018	355, 6711
LOWX	69AE	2414, 849, 850
LOWY	69A1	2403, 851, 852
LPAT	90A8	404, 405, 1114, 3941, 3985
LSBWA	890E	264, 1873, 3783, 5360, 10356
LSCALE	90A7	405, 406, 3943, 3982
LSH	BEBA	12204, 11990, 12473
LSH1	BE8D	12207
LTXMIN	0001	341, 1610, 2189
LTYMIN	0004	343, 1643, 2165
LWRCSE	0020	319, 8656
LWRFUN	0087	305, 7731
LWX005	69FB	2459, 2431
M1	891A	257, 1842, 3769, 5346, 10368
M180M	FBE4	487, 488, 3370, 3389, 6926, 7525
M2	8916	259, 1849
M360M	FBE6	486, 487, 3367, 3388, 6954
M360M2	FAE0	583, 584, 3382, 3390, 6904, 7507
MAG	FBE1	490, 491, 3348, 4334, 6866, 7001, 7863, 7882, 7894, 8197, 8208
MARG1	0002	220, 2245, 2264, 2489, 2530, 2587, 5542, 6062, 6108
MAXAP	0010	621, 8526
MAXCOL	004F	623, 9767
MAXFP	B153	9713, 9709
MAXLBL	0084	309, 542, 5032
MAXLEN	0014	612, 9575
MAXMAG	0010	304, 7884
MAXPRM	0008	313, 3086
MAXSPD	FBE3	488, 489, 3376, 6715
MAXTYP	0009	301, 3922
MDEX	BE93	12168, 11977, 12002
MEI	625B	1002, 982
MENUON	0001	639, 8739, 8757, 11014, 11128
METB	6230	980, 8890
MIDCH	0020	234, 2253, 4671, 5508
MINDX	612D	804, 794, 796
MINLEN	0002	615, 10081, 10083, 10277, 10279
MINORX	B385	10079, 10061
MINORY	B4B1	10275, 10258
MINUS	0020	245, 3052, 3068, 3098, 3107, 5577
MLEN	FFEC	359, 6528, 6568
MODSAV	FAFB	562, 566
MODSEQ	710A	3697, 690, 691
MODTAB	6112	790, 3698
MOVDN	ADC5	8944, 974, 975, 1004
MOVE	0001	178, 1128, 1300, 1305, 1321, 1678, 2368, 3656, 4184, 5311, 7923, 7939, 9910, 10910, 11153
MOVEGC	A3E6	7586, 3514, 4987
MOVLFT	ADE6	8971, 973, 1006
MOVRT	ADDF	8964, 1005, 9140

13255
 2648A MICROCODE LISTING 'GR70'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
MOVST      ADBB      8935, 978
MOVST1     ADBE      8937, 9000
MOVUP      ADD2      8954, 991, 992, 1003
MPT010     7835      4912, 4915
MPY1       7831      4909, 5454,10563
MPY45      675B      1942, 1763, 3755, 5333, 6557, 6596, 6984,10352
MPYALL     7817      4882, 4622, 4901
MPYTSZ     782E      4907, 4290, 4293, 4625, 4891
MSBD       8911      263, 7620
MSBWA      890C      265, 1875, 3784, 5361,10357
MSGON      0010      182, 7287, 7656, 7674
MUABT      AA3E      8639, 1013, 1014
MUBUF      FADA      592, 8878, 9083, 9084, 1092, 8745, 8770, 9050, 9052, 9054,
                    9056, 9058, 9060, 9062, 9064, 9066, 9068, 9070, 9072, 9074,
                    9076, 9078, 9080
MUCHK      B8EE      11126, 23, 6335, 6459, 7706, 7947, 8613, 8621, 8632, 8716,
                    8728, 8988
MUESC      AD89      8887, 965, 966, 8634
MUFLO      FB02      558, 559, 1125, 6389, 6418, 8597, 8600, 8749, 8945, 8955,
                    8982, 9183, 9292
MUI         6227      972, 964
MUKEY      0092      320, 7698, 7708
MUL        BD8C      11975, 9652, 9707, 9734, 9957,10012,10028,10154,10210,10226,
                    11584,11670,11705,11738
MULEN      024D      8877, 8878, 8771
MULP1      0009      11840,12333
MULP2      0005      11836,12335
MULP3      0001      11832,12337
MULX       BF48      12333,11980
MULX1      BF63      12350,12347
MULX2      BF66      12353,12345
MULX3      BF77      12368,12358,12364,12384
MULX4      9000      11830,12388
MULX5      BF71      12363,11841
MUOFF      AB08      8755, 8623, 8640, 8648, 8730
MUON       AAЕ3      8737, 8615, 8719
MUTB       61F8      948, 42, 8720
MUTODM     AB1F      8768, 1126
MXSTAT     000C      330, 4094
NEG        002D      329, 4415
NEGATE     A309      7372, 1587, 2035, 2044, 2070, 2073, 2449, 2464, 2470, 3100,
                    3733, 3745, 4416, 4639, 4747, 4776, 5455, 6175, 6181, 6215,
                    6218, 6543, 6584, 6970, 7403, 7427, 8477,10564
NEWCD      90D3      381, 382, 1555, 1725, 2049
NEWGCX     90CF      384, 385, 1310, 2657, 2681, 3463, 3480, 3483, 3601, 4074,
                    4197, 4536, 5060, 5717, 5719, 6490, 6663, 6669, 6883, 6901,
                    6913, 7589, 7926, 8386
NEWGCY     90CD      385, 386, 1312, 2659, 2683, 3466, 3486, 3489, 3598, 4076,
                    4195, 4538, 5062, 6493, 6671, 6687, 6885, 6924, 6934, 7592,
                    7928, 8391
NEWLIM     F88D      8878
NEWWA      0008      181, 1684, 1734, 1757, 1865, 2025, 3679, 3818, 5394, 5573,
                    5845,10380
  
```

13255

2648A MICROCODE LISTING 'GR70'
SYMBOL VALUE REFERENCED ON

13255/90010
REV 04/17/78

```

=====
NEWZM 0008 191, 7010, 7228, 7234
NEXTRM 0080 635, 9333, 9443
NIP 0001 628, 3028, 3045, 3057, 3070, 3107, 5577, 9355, 9384, 9413,
          9455, 9487, 9550, 10977
NMBO 0020 633, 9408, 9452, 9483, 9497, 9513, 9528, 9540, 9548
NMCLBF FBC3 648, 9613, 10872
NOCR 75D1 4482, 1055, 1056
NOCRLF 75C5 4472, 8670
NODC3 75CB 4475, 1059, 1060, 4486, 4512
NOFUNC 99DB 5588, 731, 732, 762, 763, 801, 802, 838, 839, 861,
          862, 897, 898, 912, 913, 936, 937, 976, 977, 1040,
          1041
NOLF 75DA 4490, 1057, 1058
NOP 99C1 5570, 758, 759, 797, 798, 811, 812, 813, 834, 835,
          932, 933, 1336
NOP1 6449 1334, 727, 728
NORM BF00 12272, 12143
NORM1 BF02 12273, 12283
NORM2 BF13 12285, 12298
NORM3 BF20 12298, 12275
NORST A220 7103, 3593, 3616, 6487, 6508, 7756
NOSOL 0004 231, 1739, 3669, 4976, 5308
NRMVEC FFFC 315, 3224, 3315
NUMBUF FB0D 542, 545, 547, 6404, 7833, 7981, 8033, 8223, 8316, 8342,
          8363, 8406, 9210, 9212, 9439, 9552, 9694, 10429, 10453, 10525,
          10532, 10547, 10659, 10735, 10747, 10754, 10798, 10819, 10820, 10845,
          10846
NUMLEN FB93 540, 541, 9557, 9573
NUMPTR FB91 541, 542, 9418, 9555, 9578, 9581, 9598
NWSIZE 6E9E 3347, 3339, 5710, 6000, 7879, 7886, 7897
NWZOOM 0080 215, 3292, 3391, 3438, 5720, 6876
OCTANT 90D1 383, 384, 1218, 1543, 1581, 1583, 1782, 2096
OCTTAB 672B 1919, 1834
OFFMNU AA29 8620, 1050
ONE 0031 325, 4182, 4344, 4369, 5939, 5956
ONEDOT 6421 1316, 738
ONEDT1 642A 1320, 1318, 1671
OP1A 0027 11871, 12413
OP1S 0019 11856, 12411
OP2A 0023 11867, 12419
OP2S 0015 11852, 12417
OP3A 001F 11863, 12425
OP3S 0011 11848, 12423
OP4A 001C 11859, 12432
OP4S 000E 11845, 12430
OP4X 002A 11874, 12434
ORGGO 7535 4352, 4151
OU BC16 11676, 10430
OUT1 BC34 11693, 11690, 11710
OUT10 BCBE 11767, 11762
OUT11 BCC0 11768, 11771
OUT12 BCC9 11773, 11784
OUT13 BCD6 11780, 11759

```

13255
 2648A MICROCODE LISTING 'GR70'

13255/90010
 REV 04/17/78

SYMBOL	VALUE	REFERENCED ON
OUT2	BC36	11694,11704,11716
OUT3	BC44	11699,11685
OUT4	BC4E	11705,11695
OUT5	BC59	11712,11697
OUT6	BC72	11724,11722
OUT7	BC7B	11733,11747,11756
OUT8	BC9C	11748,11736
OUT9	BCAF	11757,11752
OVER	002E	11878,11880,12011
OVERF	BDCA	12010,11979,11993,12003,12124,12146
OVUN	8EB6	12198,12186
P180M	FBE8	485, 486, 3364, 3385, 6979
P180M2	FADE	584, 585, 3379, 3386, 6936, 7517
P360M	FBEA	484, 485, 3361, 3384, 6914, 6963, 7499
PAGE	6A1D	2486, 28, 865, 866, 2752, 6132, 7956
PAT2	FBC7	506, 507, 9869,10109,10114,10305,10309,10376
PATENB	0004	297, 3873,10374
PATERN	8940	289, 1890, 3787, 5364,10377
PCH007	7846	4931, 5444
PCH010	786F	4963, 4933, 6200
PCH015	789D	4988, 4984
PCH017	78B9	5002, 4999
PCH018	78C7	5012, 5009
PCH019	78D4	5019, 5017
PCH020	78DD	5026, 4930
PCH050	78F4	5044, 4927, 6043, 6093, 6173, 6199, 6213, 6284
PCH1	7885	4974, 2538, 6071, 6186, 6220
PCH2	788A	4978
PENDN	63F6	1298, 736
PENGO	73DE	4175, 4144
PENORG	7344	4065, 815, 1333
PENUP	6401	1303, 735
PERIOD	8011	9451, 9346
PFBRK	0001	345, 2284, 6114
PFBUSY	0002	346, 2308, 6124
PHYSGO	7542	4362, 4152
PHYSTR	755B	4377, 4374
PINDX	60BD	734, 724, 726
PINIT	0001	360, 6348
PJMPR	0020	143, 2323, 2325, 2327
PLTESC	644F	1340, 729, 730
PLTPRM	63AE	1227, 721, 722
PLTSEQ	6391	1210, 684, 685
PLTTAB	60A2	720, 1211
PLUMIN	AJDF	9407, 9341, 9343
PLUS	002B	328, 4413
PNG010	73F7	4187, 4185
PNORG1	6443	1331, 739
PNTCNT	FB08	549, 550, 9676, 9683,10925
PNTPLT	0008	302, 1668
POINT	002E	327, 4443
PRD010	8025	9462, 9457
PRESHF	8905	274, 6995

13255
 2648A MICROCODE LISTING 'GR70'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
PREX      002F      11880,11881,11898,12148
PRMABT    AA47      8647, 995, 996
PRMBUF    90B9      388, 389, 1353, 1355, 1366, 1371, 1386, 1391, 1435, 1442,
                  1496, 1502, 1521, 1529, 3094, 3127, 3130, 3461, 3464, 3478,
                  3484, 3717, 3720, 3723, 3726, 3836, 3840, 3846, 3850, 3974,
                  3977, 3984, 4010, 4054, 4056, 5624, 5666, 5669, 5690, 5876,
                  5914, 5942, 5947, 7362, 8510, 8531, 8553
PRMCNT    6D62      3147, 3434, 3456, 3476, 3710, 3830, 3972, 4005, 4052
PRMDEX    90B6      390, 391, 1198, 1414, 1425, 1463, 1487, 1513, 1720, 3084,
                  3124, 3151, 5576, 7359, 8517, 8524
PRMSTR    A2F8      7358, 1424, 1486, 1512
PRMVEC    909C      412, 413, 1214, 1228, 1254, 1285, 1405
PTF010    AE10      9004, 8989
PTR1      FADD      585, 586, 1160, 6347
PTR2      FADC      586, 587
PTRFLG    FE77      150, 2597
PTROT1    8D20      159, 2909
PUTBRK    0005      151, 6119
PUTBUF    A9AC      8523, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026,
                  1027
PUTCHR    783B      4926, 904, 905, 1186
PUTCOL    AE89      9092, 8929, 8967, 8974
PUTFLD    ADED      8981, 6401, 8550, 8750, 8910, 8917, 8950, 8960
PYG010    7551      4371, 4368
QJMPR     0040      144, 2323, 2325
RBDWR     0010      212, 3596, 3605, 3619, 6441
RBISON    0010      201, 3591, 3594, 3614, 3617
RBO010    7076      3603, 3624
RBOFF     707E      3612, 1673, 3585, 6448, 7083
RBON      7056      3589, 7067
RBX       907F      427, 428, 3602, 3623
RBY       907D      428, 429, 3599, 3621
REL010    A0AE      6833, 6850
REL020    A0B9      6847, 6822
REL030    A0C1      6852, 6817, 6819
RELFMT    63E3      1279, 746
RELGC     A084      6806, 17
RELPRM    652D      1511, 1280
RELROW    B1E6      9841, 9773
REMOTE    0008      139
REPORT    9C03      5924
RESET     0080      247, 1076, 1199, 4367, 5551
RLFILL    71D9      3828, 810
RLP010    653D      1520
RLP020    654C      1528, 1517
RND0      BCFA      11805,11713
RNDA      BDA9      11992,11984,12008
RNDR      BF3D      12320,12313
RNGCHK    B969      11298, 9855,10881
ROND      BF2E      12309,11992,12145
RPT020    9C3E      5950, 5943, 5948
RPT030    9C41      5957, 5972
RPT040    9C51      5968, 5959
  
```


13255

2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
RSH      BEC7  12221,11530,12091
RSH0     BEC9  12222
RSH1     BECB  12223,12230
RSH2     BED8  12233,12224
RSH3     BED8  12236,12250
RSTJMP   0001   619
RSTOFF   0004   155, 7104
RSTON    0002   156, 7097
RTJUST   0010   233, 2253, 4929, 5414, 5508,10572
SANGTB   74E5   4319, 4308
SB        BDD3  12017, 8684, 8698, 9650, 9732,10026,10224,10403,11302,11308,
          11363,11368,11381,11386,11428,11433,11445,11450
SBF025   98FC   5433, 5415
SBF030   98FE   5436, 5447
SBF050   990F   5450, 5418, 5427
SCALE    90B3   393, 394, 1891, 3771, 3936, 3944, 3950, 3992
SCALER   8921   290, 1892, 3772, 5350,10371
SCLD     0040   225, 2236, 2243, 2264, 2336, 2346, 2355, 2527, 5542, 6039,
          6102
SCR       9000   369, 371,11830,11843,11861
SCRB     0090   371,11500,11512,11573,11654,11699,11755,11776,11790,11795,
          11901,11918,11928,11934,11965,12007,12010,12031,12175
SELECT   0020   129, 7032,11019
SELLED   0020   66, 7035,11022
SELWA    890B   268, 1866, 7618
SENDA    7580   4402, 4260, 4336
SENDHD   759D   4427, 4179, 4198, 4257, 4295, 4356
SENDHL   7586   4409, 4429, 4432
SENDP    75AD   4442, 4337
SENDTM   75C2   4465, 4190, 4199, 4261, 4348, 4357, 4452
SET1     6085   3180, 3171
SETBRK   690E   2283, 843
SETBSY   692D   2307, 844
SETLIN   7239   3899, 806
SETLN1   7248   3906, 3904
SETLN2   7264   3921,10934
SETMD1   7220   3869, 3867, 5661, 7967, 9867,10898
SETMEM   000A   296, 3179
SETMOD   7215   3864, 805
SETORG   732B   4050, 814
SETPF1   6910   2285, 2309
SETRTB   6386   1202, 2230, 2367, 2480, 2504, 2632, 2645, 2740, 2757, 2936,
          2968, 3161, 3699, 4085, 4248, 4474, 5596, 8721, 8901
SETSMP   0033   336, 5634, 5636
SETUP    655D   1541, 1675, 5844
SF        0035  11886,11887,12049,12070,12092,12158
SFTCR    00EF   308, 7841, 7997, 8064, 8127,11336
SFTRST   634F   1152, 13
SHL010   7592   4417, 4414
SHTFMT   6307   1267, 744
SHTPRM   64DD   1451, 1268
SIGNM1   8918   258, 1853, 3767, 5348,10369
SIGNM2   8914   260, 1856

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
SIZKEY 0097 318, 7999, 8129
SIZMSG A672 8016, 7979
SKPBF FB9E 660,10926
SKPCNT FB0A 548, 549, 9318, 9327, 9394, 9399,10928
SKPFLD 000C 606, 9295
SLANT 0001 229, 2251, 4557, 4568, 4577, 8089, 8093, 9873
SLFTST 8909 269, 1898, 7615
SLNT0 7505 4325, 4312
SLNT1 7631 4559, 4570
SLNT45 750D 4327, 4315
SLNTOF 763A 4567, 820
SLNTON 7626 4554, 819
SLOVEC FFFD 314, 3228, 3298
SLTMSG A745 8112, 8057
SNDBUF 98D1 5404, 5480
SNDCMA 75A8 4436, 4180, 4258, 4296, 4338, 4430
SNDEOT 0001 307, 2796
SNDGCF A274 7225, 3215, 3517, 3533, 6452, 7093
SNDLBO 9920 5468, 6010, 6042, 6091
SNDLBL 9928 5473,10526,10533,10554
SNDMOD 7232 3886, 3222, 3246, 7655, 7672
SNDNIL 0002 306, 2791
SNDST1 75B8 4453, 4310, 4451, 4459
SNDSTR 75B2 4450, 4168, 4268, 4278, 4317, 4375, 4384, 4393
SNDTEK 68B2 2784, 2575, 2630, 2708
SNDTKX 68D0 2806, 2786
SNDTKY 68FA 2842, 2788
SPEED FBEE 481, 482, 6718, 6777, 6792, 6848
SPF010 6927 2298, 2294
SPM010 6D3E 3106, 3082, 3087
STAPF2 B8BC 11062, 8574, 8590, 8740, 9665, 9747, 9894,10942
STAPFL B8AF 11044, 9412, 9444, 9460, 9468, 9504, 9541,10893
START B082 9547, 9415, 9461, 9489
STATGO 7389 4115, 43
STATTB 61DC 927, 4084
STATUS 7358 4082, 702, 703
STBIT 0040 192, 5853
STD010 98B1 5850, 5899
STDRAW 98A1 5843, 5756, 5780, 5806, 5830
STF010 98DB 5881, 5878
STF020 98DE 5887, 5890
STFAIL 98CA 5871, 5854
STFLAG 90A6 406, 409, 1897, 5622, 5640
STFLG1 A226 7119, 1129, 1306, 2369, 3283, 3555, 3819, 3940, 3948, 5312,
5395, 5574, 5846, 7657, 7940, 9911,10381,10920
STFLG3 A233 7138, 3572, 3595, 6489
STFLG5 A240 7155, 1324, 1687, 2661, 3213, 3250, 3293, 3392, 3439, 3469,
3574, 3597, 3620, 3909, 3997, 5397, 5721, 5848, 6690, 7594
STFLG6 A25A 7189, 2959, 4529, 4558, 4977, 5309, 8090
STFLG7 A267 7206, 1077, 1216, 1256, 3031, 3053, 3061, 9889
STFMSG 9C57 5974, 5929
STG010 73A8 4131, 4128
STGIN 6ADE 2639, 29, 867, 868

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
STGIN1    6AEA    2644, 4211
STL010    726A    3925, 3919, 3923
STL030    727F    3938, 3914
STL040    7291    3946, 3916
STM010    7228    3874, 3871
STMSG2    9C72    5978, 5933
STOP      80B4    9589, 9369, 9386, 9499, 9502, 9515, 9518, 9530, 9534, 9538,
          9627,10979
STOP2     80E0    9625, 9335, 9410, 9437, 9454, 9466
STOPAP    A9F0    8579, 1045
STOPKY    A518    7809, 7761
STOPPM    6D15    3078, 3023, 3046, 3120, 3149
STORE     80A3    9571, 9422, 9469, 9490, 9505
STORG1    733A    4057, 4069, 4077
STP010    B0DC    9617, 9615
STPRPT    0009    137, 7712, 7727,11338
STR       BD3E    11909, 8687, 8693, 8701, 8707, 9217, 9997,10015,10070,10195,
          10213,10266,10394,10397,10413,11586,11680,11775,11792
STR0      BD3C    11907,11967,12163
STR1      BD3F    11910,12318
STTKFL    A24D    7172, 2244, 2263, 2409, 4123
STVECS    73B2    4142, 4126
STX       A60C    7961, 7775
SUPCHR    0020    224, 1153, 2245, 2264, 2370, 2489, 2777, 4122, 4234
SUPRO     0001    197, 1219, 2374, 3500, 3591, 6463, 6483, 7651
SUPR1     0002    198, 3182, 3194, 3591, 3714, 3820, 6483, 8487
SUPR2     0004    199, 3231, 3257, 3591, 6483, 8487
SUPRGC    9E8A    6438, 1220, 1801, 2375, 3183, 3258, 3715, 3833, 5323, 7652
SUPRZM    0004    210, 3204, 3249, 6871
SUPTMR    FBEC    483, 484, 6449, 7024
SVAD      BCDE    11786,11550,11677
TANG      FBDB    494, 495, 4302, 4560, 4587, 4636, 4773, 4854, 5513, 6047,
          6288, 8010, 8036, 8083, 8107, 8314
TCH010    694F    2334, 2328
TCK010    BA1D    11371,11359
TCK020    BA48    11390,11377,11441
TCMSG1    BA66    11409,11403,11464
TCMSG2    BA70    11410,11405
TCX010    B32A    10020,10071
TCX020    B359    10049,10042
TCX025    B36D    10060,10057
TCX030    B370    10064,10047,10059
TCY010    B457    10218,10267
TCY020    B486    10247,10240
TCY025    B499    10257,10254
TCY030    B49C    10260,10245,10256
TEKAC     6A6F    2553, 31
TEKCP     6A81    2609, 869, 870
TEKCR     6A47    2515, 857, 858
TEKESC    6A17    2478, 859, 860
TEKHC     6C57    2933, 871, 872
TEKHOM    6A50    2524, 1079, 2493, 6111
TEKMSG    A77D    8134, 8119

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```
=====
TEKPF      FBC4      508, 509, 2291, 5544, 5561, 6113, 6123
TEKRPT     6A34     2501, 873, 874
TEKSEQ     68D1     2228, 693, 694
TEKST1     6A7A     2558, 2614
TEKTAB     614E      827, 2229
TEKTRM     FBDC      493, 494, 2278, 2790, 2795
TEMP       906B      445, 447, 455, 470
TEMPDX     9055      465, 466, 2131, 2136, 2138
TEMPDY     9053      466, 468, 2133, 2140, 2142
TESCTB     618C      864, 2479
TEXTGO     74AA     4286, 4149
TGC010     6FF0     3515, 3509
TGCOF1     7002     3531, 2655, 2692, 3522, 4236, 5537, 7823, 9862,10884
TGCOFF     6FF6     3521, 777
TGCON      6FD0     3496, 776
TGCON1     6FD6     3499, 2654, 2662, 3497, 3575, 5708, 7824, 7963
TGLAN      A5C3     7910, 7769
TGLGC      A51F     7816, 7762
TGLGVD     A5B8     7902, 7768
TGLMU      A5FA     7946, 7772
TGLRB      A573     7854, 7763
TGLZM      A57E     7862, 7764
TICCHK     B9F2     11353, 9857
TICERR     BA4E     11399, 9858
TICFLG     FAE3      576, 577, 9987,10045,10051,10185,10243,10249
TICLBL     0080      644, 9881, 9893,10056,10253
TICPTR     FAEC      573, 574, 9880, 9886, 9892, 9902, 9990, 9996,10004,10011,
10067,10188,10194,10202,10209,10263,10398
TICZRO     B557     10392,10023,10221
TIMOUT     0014      357, 6450
TIMSUP     0008      200, 1800, 3591, 5322, 6483, 7026
TINTR      A1B0     7021, 18
TKACGO     6A86     2567, 4158
TKBUSY     9D41     6127, 6129, 6131
TKCPGO     6ABF     2619, 4159
TKFLGS     90AD      399, 400, 2320, 2345, 2354, 2387, 2525, 2586, 2776, 2778,
3507, 5541, 5563, 5654, 5657, 5729, 6037, 6100, 6110, 7173,
7181, 7700
TKGCGO     6B48     2702, 4161
TKGNGO     6B42     2698, 4160
TKGSTB     616C      846, 2366, 2502, 2631, 2935
TKHU10     6A63     2534, 2532
TKHC       6C2D     2896, 30, 2934
TKINDX     6169      841, 829, 831
TKOFF      68D7     2235, 2324, 2326
TKRPT1     6A37     2503, 8642
TKSCLD     68DC     2242, 2338
TKSTAT     73CA     4157, 4129
TKSTRP     6932     2319, 41, 1078
TKSTUP     6962     2363, 27, 853, 854
TKUNSC     68F7     2261, 2332
TKX005     6BE0     2817, 2808
TKX010     6BE5     2821, 2815, 2849, 2885
=====
```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```
=====
TKY010  6C08  2850, 2844
TKY015  6C1A  2869, 2874
TKY020  6C21  2876, 2867, 2881
TMP1    0038  11889,11890,11661,11733
TMP2    0039  11890,11891,11596,11605,11700,11706,11718
TMP3    003A  11891,11892,11565,11655,11748
TMP4    003F  11896,11581
TMPBUF  90B7   389, 390, 3036, 3040, 3067, 3097
TMRINT  0003   154, 7264
TOPCH   0040   235, 2253, 4671, 4673, 5508
TOPSAV  FB00   559, 560, 8742, 8759
TRBOFF1 704E   3582, 3537, 3580, 7857
TRBOFF  7048   3579, 779
TRBON   7031   3565, 778
TRBON1  7037   3568, 3566, 7858
TRIM    AEDF   9170, 9155,10592
TRM010  AEE0   9172, 9176
TRMSTP  6900   2274, 842
TRUNCT  B93D   11235,10010,10208
TST     BD59   11934, 9947, 9952, 9992,10144,10149,10190,11358,11376,11423,
        11440,11663,11678
TST1    BD5B   11935,12039,12060,12063
TXA005  A694   8030, 8084, 8091, 8095
TXA010  A6BD   8052, 8050
TXA020  A6CD   8061, 8063, 8087
TXA025  A703   8085, 8078, 8080
TXA030  A709   8088, 8070, 8072
TXA040  A711   8092, 8074, 8076
TXA050  A719   8096, 8065, 8067
TXA060  A726   8106, 8103
TXANGL  7802   4866, 818
TXCHK   9DC6   6228, 6275
TXG010  74E2   4316, 4314
TXMAG   FBDA   495, 496, 2250, 4632, 4740, 4853, 4908, 5327, 5349, 5377,
        7984, 8007, 8340, 9876
TXORG   FBD3   499, 500, 4298, 5502
TXS005  A625   7978, 8008
TXS010  A641   7994, 7996, 8015
TXS015  A663   8009, 7998, 8000
TXS020  A66C   8013, 8002, 8004
TXSIZ1  77F1   4852, 4850, 5990
TXSIZE  77DC   4843, 817
TXT010  A765   8123, 8126, 8133
TXT1    A75B   8118, 7974, 8024
TXTANG  A686   8022, 7776
TXTSIZ  A61B   7972, 7777
UDAREA  0002   340, 3915
UDLINE  0001   339, 3913
UNSCLD  0001   219, 2236, 2245, 2262, 2330, 2346, 5542
UNZOOM  6E78   3321, 3252, 3310, 3350, 5538
USEGC   640C   1308, 737
VAL1    003C  11893,11894
VAL2    003D  11894,11895
```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
VAL3      003E  11895,11896
VALE      003B  11892,11893,11585,11594,11679,11777
VAPAT     FBEB   480,  481, 1116, 4022, 7557
VD1010    A43C   7650, 7645
VDC       8902   266, 1106, 3230, 3256, 3299, 3316
VEC010    6605   1689, 1680
VEC020    6618   1710, 1697
VEC030    6623   1717, 1688, 1702, 1709
VEC040    6647   1736, 1731
VECCNT    9051   468, 5748, 5768, 5798, 5818, 5858, 5860
VECTOR    65DB   1666, 1314, 1357, 1376, 1396, 1445, 1479, 1535, 2458, 2473,
              7930, 9673
VECTRO    65E9   1672, 1329, 1669, 5313
VECTR1    65EC   1674, 3671
VERSN     0054    3,    7, 2084, 3528, 4860, 5268, 6683, 8217, 9428,10916
VID010    A467   7681, 7678
VIDEO1    A431   7643,  38, 8738
VIDEO2    A44E   7670,  39
VLINE     B50A  10343, 9930, 9933,10082,10084,10101,10112,10118,10128,10130,
              10171
VR         A1CB   7045,  19, 3448, 7846
VR010     A1F3   7070, 7066
VR020     A1F9   7073, 7060
VR025     A211   7086, 7078, 7081
VR030     A214   7088, 7056, 7072
VRESET    8961   253, 7049, 7258
VRFLAG    0020   300, 7047, 7261
VRW010    A29D   7259, 7265, 7268
VRW020    A2B4   7269, 7262
VRWAIT    A292   7252, 1074, 3217, 3241, 3279, 3327, 3447
VSETUP    A40F   7613, 3766, 5342,10370
VT010     9AFC   5755, 5762
VT020     9B0F   5766, 5757
VT030     9B30   5779, 5786
VTEST     9AE1   5744, 5628, 5635
WAIT      A287   7240, 1100, 3206, 3253, 3297, 3314, 3322, 3534, 3812, 3887,
              5389, 5851, 5904, 6443, 6453, 6477, 6501, 6551, 6989, 7231,
              10355
WAIT15    9CAA   6011, 5706
WANTAP    0010   645, 8506, 8573, 8580, 8664, 8666
WANTAX    0040   646, 8506, 8589, 8661, 8666
WANTGC    0080   204, 3504, 3535, 4534, 4983, 5047, 7055, 7822, 8155, 8487,
              8488
WANTRB    0020   202, 3571, 3583, 7064, 7076, 7856, 8169
WANTZM    0002   209, 3209, 3251, 3292, 3308, 4342, 6713, 6871, 6872, 7867,
              7876, 7892, 8184
WAT010    A288   7242, 7245
XABS      64C7   1434
XAX       FAF9   566,  567,10141,10166,10317
XAXIS     B2B0   9942, 9871
XAXLEN    0266   617, 9922, 9931, 9972,10035,10160,10279,10306,10312,10904
XBS       9D76   6170,  35, 6313
XCD010    65B1   1614, 1607

```

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'GR70'
SYMBOL VALUE REFERENCED ON

```

=====
XCELL      0007      333, 4292
XCHADJ     9077      431, 432, 4678, 4692, 4694, 4706, 4721, 4723, 4753, 5280,
                    5337
XCHECK     A360      7478, 3462, 3482, 3718, 3724, 3838, 3842, 5285, 5288, 6668,
                    7588
XCHINC     9089      422, 423, 4964, 5417, 6174,10562
XCHSIZ     908D      420, 421, 4620, 4647, 4649, 4660, 4662, 5283
XCODE      659F      1598, 1545, 2098, 2101
XCOLBF     FBC2      649, 9605,10875
XCR        9CAF      6032, 22, 2965, 2980, 5000
XCR005     9CD4      6054, 6049
XCR010     9CDD      6060, 6040
XCR020     9CE9      6066, 6064
XCR030     9CF2      6070, 6053, 6058, 6103, 6107
XCR035     9CF5      6072, 6125, 6133, 6134
XCURR      90DE      375, 376, 1142, 1325, 1368, 1470, 1548, 1722, 1741, 2030,
                    2077, 2574, 3648, 3686, 4066, 4178, 4966, 4996, 5006, 5278,
                    5335, 5420, 5475, 5546, 5559, 5750, 5758, 5760, 5770, 5782,
                    5784, 5800, 5823, 6144, 6177, 7587, 8409,10426,10566,10568
XDEL       905D      461, 462, 2038, 2069, 2071, 2130
XDIV       8000      677, 684, 687, 690, 693, 696, 699, 702, 705, 708,
                    711, 721, 727, 729, 731, 752, 758, 762, 791, 797,
                    801, 832, 834, 838, 847, 849, 851, 853, 855, 857,
                    859, 861, 865, 867, 869, 871, 873, 878, 880, 882,
                    887, 897, 904, 906, 908, 910, 912, 916, 918, 920,
                    928, 930, 932, 936, 940, 942, 949, 951, 953, 955,
                    957, 959, 961, 965, 969, 983, 985, 987, 989, 991,
                    993, 995, 1011, 1013, 1018, 1020, 1022, 1024, 1026, 1028,
                    1030, 1036, 1038, 1040, 1055, 1057, 1059, 1061
XFB010     A9D6      8555, 8563
XFB020     A9E2      8564, 8558
XFER       AB28      8779, 4608, 4616, 8787
XFER2      AB33      8795, 6415, 8802, 9211
XFIN       9065      457, 458, 2034, 2058, 2097
XFLD1      AB79      8810, 9050
XFLD10     AC9B      8847, 9068
XFLD11     ACC7      8851, 9084, 9070
XFLD12     ACF3      8855, 9072
XFLD13     AD26      8860, 9074
XFLD14     AD4C      8865, 9076
XFLD15     AD68      8870, 9078
XFLD16     AD82      8874, 9080
XFLD2      AB94      8814, 9052
XFLD3      ABAF      8818, 9054
XFLD4      ABCA      8822, 9056
XFLD5      ABDB      8826, 9058
XFLD6      ABF6      8830, 9060
XFLD7      AC11      8834, 9062
XFLD8      AC2C      8838, 9064
XFLD9      AC6F      8843, 9083, 9066
XFRBF1     A9CD      8548, 8546, 8604
XFRBUF     A9C0      8542, 1028, 1029, 1030, 1031
XGINSV     FBD6      497, 498, 2656, 2682, 2706, 4256

```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
XHI      90AB      401, 402, 2399, 2424, 2627
XHT      9D9E      6196, 33, 6311, 8105
XINC     64F7      1468, 1499
XLBASC   F9A7      9083, 9897
XLBLBF   FBAC      656, 9891,11421,11429
XLBLFD   0008      602, 9196
XLEFT    9069      447, 448, 3719, 3742, 3779, 3839, 6536, 6545, 6961, 6972,
              7600
XLF      9CF9      6085, 36, 2973, 2985, 5001
XLF010   9D02      6092, 6323
XLF020   9D39      6121, 6115, 6118
XLF1     9E1C      6320, 6312
XLFINC   9085      424, 425, 4677, 4695, 4704, 4724, 6094, 6214
XLNG     6519      1495
XLOW     90AC      400, 401, 2417
XMARGO   0000      310, 2529, 6061
XMARG1   0103      311, 2533, 6065
XMAX     9070      435, 436, 1136, 1617, 2203,10905
XMAXBF   FB88      653, 8681,10142,11301,11360,11425
XMAXFD   0005      599, 9186
XMID     9061      459, 460, 2154, 2187, 2212
XMIN     9072      434, 435, 1133, 1602, 2192,10901
XMINBF   FBBC      652, 8683, 9649,10002,10025,10147,11299,11362,11427
XMINFD   0004      598, 9184
XMUL     0080      678, 684, 687, 690, 693, 696, 699, 702, 705, 708,
              711, 721, 727, 729, 731, 752, 758, 762, 791, 797,
              801, 832, 834, 838, 847, 849, 851, 853, 855, 857,
              859, 861, 865, 867, 869, 871, 873, 878, 880, 882,
              887, 897, 904, 906, 908, 910, 912, 916, 918, 920,
              928, 930, 932, 936, 940, 942, 949, 951, 953, 955,
              957, 959, 961, 965, 969, 983, 985, 987, 989, 991,
              993, 995, 1011, 1013, 1018, 1020, 1022, 1024, 1026, 1028,
              1030, 1036, 1038, 1040, 1055, 1057, 1059, 1061
XNEW     90DA      377, 378, 1143, 1311, 1326, 1354, 1370, 1390, 1437, 1472,
              1526, 1544, 1721, 2033, 2056, 2426, 2434, 2466, 2535, 2707,
              3638, 3644, 3690, 4537, 4968, 5013, 5061, 5422, 5486, 5665,
              5689, 5752, 5761, 5771, 5785, 5804, 5825, 6052, 6067, 6151,
              6179, 7927, 9658
XNEXT    0004      221, 2370, 2388, 2408, 2418
XOFSET   0046      613, 9655, 9920, 9924, 9927, 9931, 9971,10031,10035,10040,
              10157,10160,10163,10276,10279,10291,10294,10312,10900,10904
XORG     909A      413, 414, 1388, 1524, 2463, 2819, 3834, 4060, 4355
XPUTDC   6C60      2943, 2570, 2622, 2700, 2794, 2799, 2830, 2836
XSCALE   90A2      409, 410, 8686, 8690, 8692, 9651,10027,10153
XSOL     907B      429, 430, 1742, 6051, 6149
XSTART   9069      455, 456, 1549, 1765, 2031, 2088, 2100, 7539
XTEMP    9059      463, 464, 2127, 2152, 2213, 2221
XTICBF   FBA8      657, 9879,11356,11364
XTICFD   0009      603, 9192
XTICS    B2F6      9985, 9883, 9899
XVT      9DAA      6210, 34, 6314
YABS     64D1      1441, 1429
YAX      FAF7      567, 568, 9944, 9969,10121

```


13255
2648A MICROCODE LISTING 'GR70'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
YAXIS      B3DC    10139, 9872
YAXLEN     0130    618, 9925, 9929, 9963,10083,10110,10117,10169,10233,10906
YBOT       9067    448, 449, 3722, 3732, 3754, 3806, 3849, 6576, 6586, 7603
YBOT45     9065    449, 450, 3756, 3781, 3798, 3801
YCD010     65CF    1647, 1640
YCELL      000A    334, 4289
YCHADJ     9075    432, 433, 4680, 4698, 4700, 4709, 4728, 4730, 4754, 5293,
                    5331
YCHECK     A369    7488, 3465, 3488, 3721, 3727, 3848, 3852, 5016, 5298, 5301,
                    6677, 7591
YCHINC     9087    423, 424, 4969, 5426, 6180
YCHSIZ     9088    421, 422, 4653, 4655, 4663, 4665, 5296
YCODE      65BD    1631, 1554, 2108, 2111
YCOLBF     FBC1    650, 9609,10878
YCURR      90DC    376, 377, 1144, 1327, 1373, 1476, 1557, 1724, 1743, 2039,
                    2075, 2572, 3650, 3684, 4068, 4176, 4971, 5015, 5291, 5329,
                    5429, 5477, 5548, 5557, 5751, 5773, 5801, 5808, 5810, 5820,
                    5832, 5834, 6068, 6153, 6183, 7590, 8414,10424,10543,10545
YDEL       905B    462, 463, 2047, 2072, 2074, 2132
YFIN       9063    458, 459, 2043, 2060, 2107
YGINSV     FBD4    498, 499, 2658, 2684, 2703, 4254
YHI        90A9    403, 404, 2393, 2421, 2624
YINC       64FF    1474, 1467, 1505
YLBASC     F94F    9084, 9905
YLBLBF     FBA4    658, 9901,11438,11446
YLBLFD     000A    604, 9198
YLFINC     9083    425, 426, 4679, 4689, 4707, 4717, 6096, 6217,10541
YLANG      6523    1501, 1491
YLOW       90AA    402, 403, 2406
YMAX       906C    437, 438, 1138, 1650, 2179,10907
YMAXBF     FBB0    655, 8695, 9945,11307,11378,11442
YMAXFD     0007    601, 9190
YMID       905F    460, 461, 2158, 2214
YMIN       906E    436, 437, 1134, 1635, 2168,10903
YMINBF     FBB4    654, 8697, 9731, 9950,10200,10223,11305,11380,11444
YMINFD     0006    600, 9188
YNEW       90D8    378, 379, 1145, 1313, 1328, 1356, 1375, 1395, 1444, 1478,
                    1534, 1552, 1723, 2042, 2054, 2423, 2436, 2457, 2467, 2472,
                    2537, 2704, 3640, 3646, 3688, 4539, 4973, 5020, 5063, 5431,
                    5484, 5668, 5692, 5754, 5775, 5802, 5811, 5821, 5835, 6057,
                    6069, 6160, 6185, 7929, 9740
YOFSET     002D    614, 9737, 9921, 9925, 9928, 9932, 9960, 9963, 9966,10080,
                    10083,10096,10099,10111,10117,10168,10229,10233,10238,10902,
                    10906
YORG       9098    414, 415, 1393, 1532, 2469, 2847, 3844, 4058, 4353
YSCALE     909E    410, 412, 8700, 8704, 8706, 9733, 9956,10225
YSOL       9079    430, 431, 1744, 6056, 6158
YSTART     9067    456, 457, 1558, 1762, 2040, 2090, 2110, 7542
YTEKHM     015E    312, 2536
YTEMP      9057    464, 465, 2129, 2156, 2215, 2219
YTICBF     FBA0    659, 9885,11374,11382
YTICFD     000B    605, 9194
YTICS      B423    10183, 9887, 9907

```

13255
 2648A MICROCODE LISTING 'GR70'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
ZAH1      8910      278, 6990
ZALO      8912      277, 6991
ZB2DA     00CC      86, 4404
ZB2DDE    00CF      87, 4422
ZBELL     4814      56, 8014, 8086
ZBLFLG    FF0E      65, 7034, 11025
ZBNDCA    00AB      89, 5995, 8227
ZBNDEC    00A8      88, 8480
ZBRK1     6800      2083, 3526, 2085
ZBRK1C    6802      2086, 2081
ZBRK2     7000      3527, 4858, 3529
ZBRK3     7800      4859, 4861
ZBRK4     9800      5267, 6681, 5269
ZBRK5     A000      6682, 8215, 6684
ZBRK5C    A002      6685, 6680
ZBRK6     A800      8216, 9426, 8218
ZBRK6C    A802      8219, 8214
ZBRK7     B000      9427, 10914, 9429
ZBRK7C    B002      9430, 9425
ZBRK8     B800      10915, 10917
ZBRK8C    B802      10918, 10913
ZCAFLG    FF67      149, 4960
ZCHAR     FF88      107, 1452, 2391, 2397, 2404, 2415, 2505, 2686, 2748, 3009,
          3049, 4509, 5580, 7364, 8533, 8543, 8655, 9121
ZCHECK    A372      7497, 6950
ZCHINT    0082      74, 2507, 4511
ZCHKSF    00C6      83, 1177, 6242, 6461, 7717, 8374, 8428, 10868
ZCHRIN    FF9C      145, 2731, 4240
ZCK010    A38B      7513, 7503
ZCK020    A3A7      7530, 7522
ZCKCTL    00F0      100, 968, 2672, 4219
ZCKRMT    00D2      90, 2554, 2610, 2640, 4092, 6117, 8376, 8430
ZCLBXF    00D8      92, 4121
ZCLMD1    4811      60
ZCRADV    00BD      78, 1206, 4548
ZCRRET    00C0      79, 2518, 2768, 4485, 8494, 9383, 9387
ZCTCOL    FF12      62, 7818
ZCTLAL    9214      170, 171
ZCTLKY    0001      63, 7819
ZCTMON    282F      158, 7630
ZCURCL    FFC1      117, 8922, 8938, 8965, 8972, 9112, 9130
ZCUROW    FFC0      112, 3548, 8743, 8762, 8993, 9560, 9759, 9762, 9779, 9812,
          9818, 10966, 10970, 10973, 10981, 10992
ZDCCTL    5011      152, 6120
ZDCHAR    FF89      124, 4935, 5036, 5441, 9316, 9629
ZDCIO     00C3      82, 2671, 2727, 2746, 2763, 4218, 4230, 4483, 4491, 5925,
          9126, 11183
ZDELAY    00B4      75, 2946, 3188, 6013
ZDPMG2    00ED      99, 7837
ZDS2BF    0020      132, 10948, 10999
ZDSPC0    00BA      77, 9770, 9782, 9814
ZDSPCH    00C9      85, 11187
ZDSPLM    FBFF      119, 479, 1091, 1092
  
```

13255

2648A MICROCODE LISTING 'GR70'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
ZDSPMG 0040 71, 5932, 5965, 7992, 8060, 8122,11332
ZDSPST FFFE 123, 8746
ZENTER 00E7 97, 998
ZEOP 00CE 116, 3278, 5941, 5977, 7836, 7990, 8056, 8135, 8876,11345,
11410,11470
ZERO 0030 324, 4186, 4340, 4365
ZESCAP 00B7 76, 761, 800, 837, 894, 935, 1342, 2993, 4237, 8633
ZESCFG FFD1 108, 1204, 2508, 4476, 8888,11162
ZESCND 004F 73, 715, 896, 1000, 2382, 2488, 2495, 2516, 2556, 2562,
2612, 2642, 2693, 2719, 2897, 2983, 4508, 5584, 5618, 8732,
10909
ZGETAL 920E 168, 169
ZGETDP 0088 80,10968
ZGETKY 4805 59, 7632
ZHANG 00EA 98, 5949
ZIN010 A5A0 7881, 7877
ZIN2AL 9205 165, 166
ZINIAL 9202 164, 165
ZINITG 00E4 96,10945
ZINKEY 008B 321, 7722
ZINTAL 9208 166, 167
ZIOCCL 8700 122, 8939, 9113
ZIOCRW 8720 114, 3281, 3557, 7298, 7680, 8994
ZIOFL2 FF64 131,10947,10998
ZIOINP FFD9 127
ZIORGO 00A5 102, 2904, 7631
ZKBCTL 4808 61, 7713, 7728,11339
ZKBLED FF0C 64, 6262,11029
ZKBTMR 91EC 58, 6844
ZLCHAR FF69 113
ZLCLN2 00E1 95, 8458
ZLFTMG FFBF 130, 9798
ZLNFD 008B 81, 4492, 8495, 9397, 9400,10982
ZLWASC 482C 57, 6811
ZMAIN 0040 70, 71, 72, 73, 74
ZMDFL1 FFF4 128, 6253, 7031,11018
ZMDFL2 FFF3 138
ZMG010 752E 4345, 4343
ZMONAL 920B 167, 168
ZMOUT A5AA 7890, 7766
ZMPT1 FFF1 110
ZMPT2 FFEF 111
ZMSGAL 921A 172
ZMSGP1 FFF1 133, 5930, 5934, 7832, 7980, 8032, 8120, 8443, 8444,11324,
11402,11463
ZMSGP2 FFEF 134, 5937, 7834, 7982, 8034, 8224, 8317, 8343, 8362, 8405,
8440,11326,11404,11465
ZMSGP3 FFEF 135, 8058, 8364, 8407,11322,11400,11461
ZMSGP4 FFEF 136, 8047,11328,11406,11467
ZMT010 9CA3 6007, 6005
ZMTST 9C80 5987, 5675, 5678, 5681, 5684
ZMUPDA A0CA 6864, 3214, 7090
ZMXROW 0017 115, 3280, 3556, 7679

```

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'GR70'

SYMBOL VALUE REFERENCED ON

```

=====
ZOFF      6E58      3304, 773
ZOFF1     6E5E      3307, 3305, 5686, 7865, 7868
ZON       6E3B      3288, 772
ZON1      6E41      3291, 3289, 5672, 5711, 7869, 7880
ZOOM      0002      190, 3226, 3324, 7010
ZOOMGO    7515      4332, 4150
ZOOMIN    A590      7873, 7765
ZOOMRC    8908      276, 7006
ZOOMTB    6EF0      3394, 3358
ZOOMWC    8906      275, 6997
ZOUTKY    008C      322, 7724
ZPOS      6F68      3432, 775
ZPOS1     6F73      3437, 3435, 5671, 5694
ZPRMSG    00DE      94, 8649
ZPUTAL    9211      169, 170, 2903
ZPUTDC    007C      84, 2948, 4189, 4300, 4347, 4373, 4403, 4419, 4421, 4438,
          4444, 4457
ZRNGTA    FFD2      109, 1203, 1212, 2955, 8505, 8891
ZRO       BD46      11918,10411,11938,11952,11978
ZRO1      BD48      11919,12004,12144
ZRSTDP    0043      72, 3269, 7842, 7844, 8012, 8109, 8128, 8130, 8132, 8763,
          11340
ZSBXFA    005B      101, 2679, 2718
ZSBXFR    00D5      91, 2561, 4104
ZSDTRM    00DB      93, 4466
ZSIZE     6E89      3333, 774
ZSTAAL    9217      171, 172
ZTLINO    FFA3      126, 9830, 9842
ZTOPLN    FFCB      125, 8741, 8748, 8760
ZUD010    A0F2      6891, 6878
ZUD020    A11C      6920, 6910
ZUD030    A140      6943, 6931
ZUD040    A143      6949, 6887
ZX        FBDF      491, 492, 6884, 6908, 6921, 6956, 6965, 7501, 7514
ZXTEMP    905D      453, 6958, 6986
ZY        FBDD      492, 493, 6886, 6929, 6944, 6981, 7520, 7531
1482 SYMBOLS, 6339 REFERENCES, -62 WORK TRACKS

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
1	0000	.	ASB,HEX ;I0273 26MAY77	1
2	0000	.	*****	
3	0054	.	VERSN EQU 1240 ;GRAPHICS VERSION = 'T'	
4	0000	.	; THIS IS I0270 MODIFIED FOR GRAPHICS	
5	0000	.	; CHANGES ARE AS FOLLOWS	
6	0000	.	; 1. WHEN READING BUFFER TO DISPLAY IN TEK MODE,	
7	0000	.	; CR/LF ARE NOT STRIPPED, AND NO CR/LF IS	
8	0000	.	; APPENDED.	
9	0000	.	; 2. WHEN GOING DISPLAY TO BUFFER, A TEST IS MADE	
10	0000	.	; FOR AUTO PLOT RECORD.	
11	0000	.	; 3. CLEAR TO SEND LINE IS NO LONGER MONITORED	
12	0000	.	; FOR RS-232 PRINTER.	
13	0000	.	; 4. IN PRINTER DRIVER, A NULL CONFIGURATION OF	
14	0000	.	; 56 (ALL STRAPS CLOSED) NOW GENERATES 0	
15	0000	.	; NULLS.	
16	0000	.	; 5. I/O BUFFER IS FLUSHED WHEN RECORD KEY TURNED	
17	0000	.	; OFF.	
18	0000	.	; 6. REFERENCES TO RETURN KEY CODE (15B) HAVE BEEN	
19	0000	.	; BEEN REPLACED BY SOFT RETURN CODE (357B).	
20	0000	.	; 7. ALTERNATE I/O MOVED TO 36.5 K (111000B).	
21	0000	.	; 8. SUBROUTINE RETSCN MOVED TO KEYBOARD CODE	
22	0000	.	; TO MAKE ROOM FOR THESE PATCHES.	
23	0000	.	*****	
24	0000	.	;	
25	0000	.	*****	
26	0000	.	; SYMBOLS USED BY GRAPHICS PATCHES	
27	482E	.	RETSCN EQU 44056Q ;NOW IN KEYBOARD CODE	
28	4831	.	RETSC0 EQU RETSCN+3	
29	4834	.	USRINT EQU RETSC0+3	
30	00EF	.	SFTCR EQU 357Q	
31	6061	.	ZGGTST EQU 60141Q ;TEST FOR GRAPHICS GET	
32	6070	.	ZCHKTK EQU 60160Q ;TEST FOR TEK MODE ON	
33	0000	.	*****	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
35	0000	. . .	;	2
36	0000	. . .	; COMMON EQUATES - CM34 - 6/10/76 - 1315 HRS.	
37	0000	. . .	;	
38	9100	. . .	FSTRAM EQU 110400Q ;FAST RAM LOWER LIMIT	
39	0000	. . .	;*****	
40	0000	. . .	; KBDCSW - KEYBOARD DATA COMM SWITCHES *	
41	0000	. . .	;*****	
42	0080	. . .	FULDUP EQU 200Q ;HALF/FULL DUPLEX	
43	0000	. . .	;*****	
44	0000	. . .	; KBJMPK - KEYBOARD INTERFACE JUMPERS *	
45	0000	. . .	;*****	
46	0000	. . .	;	
47	0000	. . .	; JUMPERS SENSED AS 0' WHEN INSERTED	
48	0000	. . .	;	
49	0000	. . .	; ALL JUMPERS ARE NORMALLY INSERTED	
50	0000	. . .	;	
51	0001	. . .	CONDIS EQU 001Q ;CONTROL CODE DISABLE	
52	0000	. . .	(0=DISABLED)	
53	0002	. . .	SPLDIS EQU 002Q ;SPOW LATCH DISABLE	
54	0000	. . .	(0=DISABLED)	
55	0004	. . .	LINWRP EQU 004Q ;COLUMN 80 AUTO CR,LF	
56	0000	. . .	(0=ENABLED)	
57	0008	. . .	PAGSTR EQU 010Q ;PAGE MODE STRAP	
58	0000	. . .	(0=LINE-FIELD MODE)	
59	0010	. . .	LFPOS EQU 20Q ;LINE FEED POSITION	
60	0000	. . .	(0 = POSITION LINE FEED	
61	0000	. . .	AT START OF NEXT I/O	
62	0000	. . .	READ	
63	0000	. . .	1 = PUT LINE FEED AT END	
64	0000	. . .	OF RECORD)	
65	0020	. . .	FSTSND EQU 40Q ;9600 BAUD DATACOM SHIFT	
66	0000	. . .	(0=9600 BAUD FOR ESC,E)	
67	0040	. . .	HNDSHK EQU 100Q ;BLOCK TRANSFER HANDSHAKE	
68	0000	. . .	(0 = FOLLOW DC2SND SETTING	
69	0000	. . .	1 = SEND DC2 BEFORE DATA)	
70	0080	. . .	DC2SND EQU 200Q	
71	0000	. . .	(0 = SEND DC2 ON ENTER	
72	0000	. . .	AND FUNCTION KEY IN	
73	0000	. . .	BLUCK MODE	
74	0000	. . .	1 = INHIBIT ALL DC2	
75	0000	. . .	HANDSHAKE)	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 3
=====
 77      0000      . . .      ;*****
 78      0000      . . .      ; KBJMP2 - SECOND SET OF KEYBOARD JUMPERS *
 79      0000      . . .      ;*****
 80      0001      . . .      AUTTRM EQU 1Q          ;AUTO TERMINATE ON "ENTER"
 81      0002      . . .      CLRTRM EQU 2Q          ;CLEAR TERMINATOR ON TRANSMI
 82      0004      . . .      NOTEST EQU 4Q          ;INHIBIT TERMINAL SELF-TEST
 83      0008      . . .      EDTWRP EQU 10Q         ;INVERT SENSE OF EDIT WRAP
 84      0010      . . .      PRNTAL EQU 20Q         ;SEND ALL CODES TO PRINTER
 85      0080      . . .      DCJMP0 EQU 200Q        ;DATA COMM JUMPER
 86      0000      . . .      ;*****
 87      0000      . . .      ; KBJMP3 - THIRD SET OF KEYBOARD JUMPERS *
 88      0000      . . .      ;*****
 89      0001      . . .      DCJMP1 EQU 1Q          ;DATA COMM JUMPERS
 90      0002      . . .      DCJMP2 EQU 2Q          ;.
 91      0004      . . .      DCJMP3 EQU 4Q          ;.
 92      0008      . . .      DCJMP4 EQU 10Q         ;.
 93      0010      . . .      NODCST EQU 20Q         ;INHIBIT DATA COMM SELF-TEST
 94      0000      . . .      ; (0 = DISABLED)
 95      0020      . . .      SETCH EQU 40Q          ;TURN ON "CH" CONTROL LINE
 96      0000      . . .      ; (0 = OFF, 1 = ON)
 97      0040      . . .      CHEKCC EQU 100Q        ;MONITOR CC CONTROL LINE
 98      0000      . . .      ; (1 = ENABLED)
 99      0080      . . .      FRCPTY EQU 200Q        ;FORCE PARITY ON/NO IN CHECK
100      0000      . . .      ; (1 = ENABLED)
101      0000      . . .      ;*****
102      0000      . . .      ; CMFLGS - COMMON FLAGS *
103      0000      . . .      ;*****
104      0001      . . .      BLKTRG EQU 1Q          ;BLUCK TRANSFER TRIGGER
105      0002      . . .      INSWRP EQU 2Q          ;INSERT WITH WRAP AROUND
106      0004      . . .      FRCRST EQU 4Q          ;FORCE FULL TERMINAL RESET
107      0008      . . .      DEFSKY EQU 10Q         ;DEFINE SOFT KEY MODE ENABLE
108      0010      . . .      REMSET EQU 20Q         ;REMOTE MODE ENABLED
109      0020      . . .      RCVMD E EQU 40Q        ;TERMINAL IN RECEIVE MODE
110      0000      . . .      ;*****
111      0000      . . .      ; ERRFLG - ERROR FLAGS *
112      0000      . . .      ;*****
113      0001      . . .      DCMERR EQU 1Q          ;DATACOM (1 = ERROR)
114      0002      . . .      TESTOK EQU 2Q          ;SELF-TEST (0 = ERROR)
115      0004      . . .      LDRCHK EQU 4Q          ;LOADER CHECKSUM (0 = ERROR)
116      0000      . . .      ;*****
117      0000      . . .      ; INTFLG - INTERRUPT FLAG *
118      0000      . . .      ;*****
119      0003      . . .      TMRINT EQU 3           ;TIMER INTERRUPT
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE   4
=====
121      0000      . . .      ;*****
122      0000      . . .      ; PRCCTL - PROCESSOR CONTROL FLAGS *
123      0000      . . .      ;*****
124      0000      . . .      TMIACK EQU 0Q          ;ACKNOWLEDGE TIMER INTERRUPT
125      0000      . . .      ;              (BIT 1 OFF)
126      0001      . . .      TMRON EQU 1Q          ;SET TIMER ON
127      0002      . . .      TMIEN EQU 2Q          ;RE-ENABLE TIMER INTERRUPT
128      0010      . . .      DCIOFF EQU 20Q        ;DISABLE DATA COMM INTERRUPT
129      0020      . . .      TMIOFF EQU 40Q        ;DISABLE TIMER INTERRUPTS
130      0040      . . .      POLL EQU 100Q         ;POLL CTU INTERRUPTS
131      0000      . . .      ;V*V*V*V* SET TO ZERO FOR ROM VERSION *V*V*V*V*
132      0080      . . .      SETROM EQU 200Q       ;DISABLE (1)/ENABLE (0) ROM
133      0000      . . .      ;*****
134      0000      . . .      ; MDFLG1 - TERMINAL MODE FLAGS 1 *
135      0000      . . .      ;*****
136      0001      . . .      DSPFNC EQU 1Q         ;DISPLAY FUNCTIONS ENABLED
137      0002      . . .      INSCHR EQU 2Q         ;INSERT CHARACTER ENABLED
138      0004      . . .      MEMLOK EQU 4Q         ;MEMORY LOCK ENABLED
139      0008      . . .      FORMAT EQU 10Q        ;FORMAT MODE ENABLED
140      0010      . . .      EDIT EQU 20Q          ;EDIT MODE ENABLED
141      0020      . . .      SELECT EQU 40Q        ;SELECT MODE ENABLED
142      0040      . . .      RECORD EQU 100Q       ;RECORD MODE ENABLED
143      0080      . . .      FORGN EQU 200Q        ;FOREIGN MODE ENABLED
144      0000      . . .      ;*****
145      0000      . . .      ; MDFLG2 - TERMINAL MODE FLAGS 2 *
146      0000      . . .      ;*****
147      0001      . . .      CAPSLK EQU 1Q         ;CAPS LOCK ENABLED
148      0002      . . .      BLKMDE EQU 2Q         ;BLOCK MODE ENABLED
149      0004      . . .      AUTOLF EQU 4Q         ;AUTO LF ENABLED
150      0008      . . .      REMOTE EQU 10Q        ;REMOTE ENABLED
151      0020      . . .      WBSR EQU 40Q          ;WRITE-BACKSPACE-READ MODE
152      0000      . . .      ;*****
153      0000      . . .      ; RADIX - BASE OF INPUT PARAMETER FOR ESC SEQ *
154      0000      . . .      ;*****
155      000A      . . .      DECRDX EQU 10         ;DECIMAL NUMBERS
156      0008      . . .      OCTRDX EQU 8          ;OCTAL NUMBERS
=====

```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	5
158	0000	.	. ;*****		
159	0000	.	. ; COMMON VARIABLES *		
160	0000	.	. ;*****		
161	9165	.	. INTVEC EQU FSTRAM+1450 ;CENTRAL INTERRUPT VECTOR		
162	9168	.	. SCNVEC EQU INTVEC+3 ;FOREIGN TERMINAL DISPLY SCA		
163	0000	.	. ;		
164	FFFF	.	. COMMON EQU 1777770 ;UPPER LIMIT OF COMMON AREA		
165	00FF	.	. CMBASE EQU COMMON/256 ;MSB OF COMMON ADDRESSES		
166	FF00	.	. CMSTOR EQU CMBASE*256 ;MSB ADJUSTMENT FACTOR		
167	0000	.	. ;		
168	FFFE	.	. DISPST EQU COMMON-1 ;DISPLAY REFRESH START PTR		
169	FFFD	.	. TRMTYP EQU DISPST-1 ;TERMINAL TYPE NUMBER		
170	FFFC	.	. KBDCSW EQU TRMTYP-1 ;KEYBOARD DATACOM SWITCHES		
171	FFFB	.	. KBJMPR EQU KBDCSW-1 ;KEYBOARD STRAPS		
172	FFFA	.	. KBJMP2 EQU KBJMPR-1 ;SET 2		
173	FFF9	.	. KBJMP3 EQU KBJMP2-1 ;SET 3		
174	FFF8	.	. CMFLGS EQU KBJMP3-1 ;COMMON FLAGS		
175	FFF7	.	. ERRFLG EQU CMFLGS-1 ;ERROR FLAGS		
176	FFF6	.	. INTFLG EQU ERRFLG-1 ;INTERRUPT FLAG		
177	FFF5	.	. PRCCTL EQU INTFLG-1 ;PROCESSOR CONTROL FLAGS		
178	FFF4	.	. MDFLG1 EQU PRCCTL-1 ;TERMINAL MODE FLAGS 1		
179	FFF3	.	. MDFLG2 EQU MDFLG1-1 ;AND 2		
180	FFF1	.	. MSGPT1 EQU MDFLG2-2 ;MESSAGE POINTERS		
181	FFEF	.	. MSGPT2 EQU MSGPT1-2 ;.		
182	FFED	.	. MSGPT3 EQU MSGPT2-2 ;.		
183	FFEB	.	. MSGPT4 EQU MSGPT3-2 ;.		
184	FFE9	.	. MSGPT5 EQU MSGPT4-2 ;.		
185	FFE7	.	. MSGPT6 EQU MSGPT5-2 ;.		
186	FFE5	.	. MSGPT7 EQU MSGPT6-2 ;.		
187	FFE3	.	. MSGPT8 EQU MSGPT7-2 ;.		
188	FFE1	.	. CTIVEC EQU MSGPT8-2 ;CTU INTERRUPT VECTOR		
189	FFE0	.	. CTIJMP EQU CTIVEC-1 ;JUMP CODE FOR VECTOR		
190	FFDE	.	. IODATA EQU CTIJMP-2 ;ESQ SEQ PARM ACCUMULATOR		
191	FFDD	.	. IOCSGN EQU IODATA-1 ;SIGN FOR PARAMETER		
192	FFDC	.	. IOPSGN EQU IOCSGN-1 ;PARAMETER SIGN		
193	FFD8	.	. PARM1 EQU IOPSGN-1 ;ESCAPE SEQUENCE PARAMETERS		
194	FFDA	.	. PARM2 EQU PARM1-1 ;.		
195	FFD9	.	. PARM3 EQU PARM2-1 ;.		
196	FFD8	.	. PARM4 EQU PARM3-1 ;.		
197	FFD7	.	. PARM5 EQU PARM4-1 ;.		
198	FFD5	.	. PARM6 EQU PARM5-2 ;.		
199	FFD4	.	. RADIX EQU PARM6-1 ;RADIX OF PARAMETERS		
200	FFD2	.	. RNGTA EQU RADIX-2 ;CHAR FUNCTION TABLE ADDRESS		
201	FFD1	.	. ESCFLG EQU RNGTA-1 ;ESCAPE SEQUENCE FLAG		
202	0000	.	. ; = 0, NOT IN ESCAPE SEQ		
203	0000	.	. ; # 0, ESC SEQ IN PROGRESS		
204	FFD0	.	. RSTTMR EQU ESCFLG-1 ;SOFT RESET TIMER		
205	0000	.	. ; * * * * *		
206	0000	.	. ; END OF COMMON EQUATES		
207	0000	.	. ;*****		

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
209	0000	. . .	;*****	6
210	0000	. . .	; KEYBOARD ENTRY VECTOR POINTERS *	
211	0000	. . .	;*****	
212	4800	. . .	ZKBBAS EQU 440000 ;KEYBOARD START ADDRESS	
213	4802	. . .	ZINIKB EQU ZKBBAS+2 ;INITIALIZE KEYBOARD	
214	4805	. . .	ZGETKY EQU ZINIKB+3 ;GET KEYBOARD KEY	
215	4808	. . .	ZKBCTL EQU ZGETKY+3 ;PERFORM KEYBOARD CONTROL	
216	4808	. . .	ZKBMON EQU ZKBCTL+3 ;MONITOR KEYBOARD	
217	480E	. . .	ZSTMD1 EQU ZKBMON+3 ;SET MODE 1 FLAGS	
218	4811	. . .	ZCLMD1 EQU ZSTMD1+3 ;CLEAR MODE 1 FLAGS	
219	4814	. . .	ZBELL EQU ZCLMD1+3 ;SOUND THE BELL	
220	4817	. . .	ZSTXMT EQU ZBELL+3 ;TURN ON TRANSMIT LED	
221	481A	. . .	ZCLXMT EQU ZSTXMT+3 ;TURN OFF TRANSMIT LED	
222	481D	. . .	ZSTJPR EQU ZCLXMT+3 ;SET JUMPERS ESC SEQ ROUTINE	
223	4820	. . .	ZSTLKY EQU ZSTJPR+3 ;SET LATCHING KEYS ROUTINE	
224	4823	. . .	ZALPCK EQU ZSTLKY+3 ;ALPHA KEY ENTRY CHECK	
225	4826	. . .	ZNUMCK EQU ZALPCK+3 ;NUMERIC KEY ENTRY CHECK	
226	0000	. . .	;	
227	0000	. . .	; KEYBOARD CONSTANTS	
228	0000	. . .	;	
229	4829	. . .	FRSALT EQU ZNUMCK+3 ;INITIAL ALTERNATE CHAR SET	
230	482A	. . .	ALTOUT EQU FRSALT+1 ;INITIAL ALTERNATE CHAR OUT	
231	0000	. . .	;	
232	0000	. . .	; KEYBOARD CONTROL CALLS	
233	0000	. . .	;	
234	0001	. . .	LOCKKB EQU 1 ;LOCK KEYBOARD	
235	0002	. . .	UNLKB EQU 2 ;UNLOCK KEYBOARD	
236	0003	. . .	RPTKEY EQU 3 ;REPEAT LAST KEY HIT	
237	0004	. . .	STBLMD EQU 4 ;SET PERMANENT BLOCK MODE	
238	0005	. . .	STRTST EQU 5 ;START SELF-TEST	
239	0006	. . .	ENDTST EQU 6 ;END SELF-TEST	
240	0007	. . .	RSETKB EQU 7 ;RESET KEYBOARD	
241	0008	. . .	CKIOKY EQU 8 ;CHECK FOR I/O CONTROL KEY	
242	0009	. . .	STPRPT EQU 9 ;STUP KEY REPEAT	
243	000A	. . .	CKBRKY EQU 10 ;CHECK FOR BREAK KEY DOWN	
244	000B	. . .	SWCHAR EQU 11 ;SWITCH CHARACTER SET	
245	000C	. . .	SETFRN EQU 12 ;UPDATE FOREIGN MODE	
246	000D	. . .	STCHST EQU 13 ;SET FOREIGN OUTPUT MODE	
247	000E	. . .	FRNMD1 EQU 14 ;SET FOREIGN MODE 1	
248	000F	. . .	FRNMD2 EQU 15 ;SET FOREIGN MODE 2	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
250	0000	. . .	;*****	7
251	0000	. . .	;	
252	0000	. . .	; DATACOM CONSTANTS	
253	0000	. . .	;	
254	0000	. . .	;*****	
255	5000	. . .	ZDCBAS EQU 50000 ;DATACOM START ADDRESS	
256	5002	. . .	TRIGGR EQU ZDCBAS+2 ;BLOCK TRANSFER TRIGGER	
257	5003	. . .	RECSEP EQU TRIGGR+1 ;RECORD SEPARATOR CHARACTER	
258	5004	. . .	BLKTRM EQU RECSEP+1 ;BLOCK TERMINATOR CHARACTER	
259	5005	. . .	DCJMSK EQU BLKTRM+1 ;DATA COMM JUMPER MASK	
260	5006	. . .	DCJMS2 EQU DCJMSK+1 ;DATA COMM JUMPER MASK #2	
261	0000	. . .	;*****	
262	0000	. . .	;	
263	0000	. . .	; DATACOM ENTRY VECTOR POINTERS	
264	0000	. . .	;	
265	0000	. . .	;*****	
266	5008	. . .	ZINIDC EQU ZDCBAS+100 ;INITIALIZE DATACOM	
267	5008	. . .	ZIN2DC EQU ZINIDC+3 ;INITIALIZATION CONTINUATOR	
268	500E	. . .	ZDCMON EQU ZIN2DC+3 ;MONITORING ROUTINE	
269	5011	. . .	ZDCCTL EQU ZDCMON+3 ;MISC CONTROL FUNCTIONS	
270	5014	. . .	ZDCTST EQU ZDCCTL+3 ;SELF-TEST	
271	5017	. . .	ZGETDC EQU ZDCTST+3 ;GET DC CHARACTER	
272	501A	. . .	ZPUTDC EQU ZGETDC+3 ;PUT DC CHARACTER	
273	501D	. . .	ZGTBIN EQU ZPUTDC+3 ;GET BINARY DC CHARACTER	
274	5020	. . .	ZSTBIN EQU ZGTBIN+3 ;START BINARY OUTPUT	
275	5023	. . .	ZNDBIN EQU ZSTBIN+3 ;END BINARY OUTPUT	
276	5026	. . .	ZDCINT EQU ZNDBIN+3 ;DATACOM INTERRUPTS	
277	0000	. . .	;*****	
278	0000	. . .	;	
279	0000	. . .	; DATACOM CONTROL CALL CODES	
280	0000	. . .	;	
281	0000	. . .	;*****	
282	0000	. . .	CLRTRG EQU 0 ;CLEAR BLOCK TRANSFER TRIGGE	
283	0001	. . .	SETTRG EQU 1 ;SET BLOCK TRANSFER TRIGGER	
284	0002	. . .	RSETDC EQU 2 ;RESET DATACOM	
285	0003	. . .	SETREM EQU 3 ;SET REMOTE MODE	
286	0004	. . .	SETLCL EQU 4 ;SET LOCAL MODE	
287	0005	. . .	PUTBRK EQU 5 ;OUTPUT BREAK SIGNAL	
288	0006	. . .	DISCNT EQU 6 ;MODEM DISCONNECT	
289	0007	. . .	ENDBLK EQU 7 ;TERMINATE OUTPUT MESSAGE	
290	0008	. . .	SETMON EQU 8 ;ENTER MONITOR MODE	
291	0009	. . .	SETNRM EQU 9 ;ENTER NORMAL MODE	
292	000A	. . .	FSTBIN EQU 10 ;ENTER FAST BINARY OUT MODE	
293	000B	. . .	SNDATN EQU 11 ;SEND ATTENTION CODE	
294	000C	. . .	SNDFCT EQU 12 ;SEND FUNCTION DATA	
295	000D	. . .	PROMPT EQU 13 ;SEND PROMPT CODE	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE   8
=====
297      0000      . . .      ;*****
298      0000      . . .      ; ALTERNATE I/O ENTRY VECTORS *
299      0000      . . .      ;*****
300      0000      . . .      ;*****
301      9200      . . .      ALTORG EQU 111000Q ;ALTERNATE I/O = 36.5 K
302      0000      . . .      ;*****
303      9202      . . .      ZINIAL EQU ALTORG+2 ;INITIALIZATION ROUTINE
304      9205      . . .      ZIN2AL EQU ZINIAL+3 ;INITIALIZATION CONTINUATOR
305      9208      . . .      ZINTAL EQU ZIN2AL+3 ;INTERRUPT PROCESSOR
306      920B      . . .      ZMONAL EQU ZINTAL+3 ;MONITORING ROUTINE
307      920E      . . .      ZGETAL EQU ZMONAL+3 ;INPUT ROUTINE
308      9211      . . .      ZPUTAL EQU ZGETAL+3 ;OUTPUT ROUTINE
309      9214      . . .      ZCTLAL EQU ZPUTAL+3 ;CONTROL ROUTINE
310      9217      . . .      ZSTAAL EQU ZCTLAL+3 ;STATUS ROUTINE
311      921A      . . .      ZMSGAL EQU ZSTAAL+3 ;ALTERNATE DEVICE NAME
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
313	0000	.	.	;*****	9
314	0000	.	.	; ASCII CHARACTER EQUATES *	
315	0000	.	.	;*****	
316	0000	.	.	NULL EQU 00 ;NULL	
317	000A	.	.	LF EQU 120 ;LINE FEED	
318	000C	.	.	FF EQU 140 ;FORM FEED	
319	000D	.	.	CR EQU 150 ;RETURN	
320	000E	.	.	SO EQU 0160	
321	000F	.	.	SI EQU 0170	
322	0012	.	.	DC2 EQU 220 ;DEVICE CONTROL 2	
323	0013	.	.	DC3 EQU 230 ;DEVICE CONTROL 3	
324	0018	.	.	ESC EQU 330 ;ESCAPE	
325	0020	.	.	CTLLIM EQU 400 ;CONTROL CODE UPPER LIMIT	
326	0020	.	.	ABLNK EQU 0400 ;ASCII BLANK	
327	0026	.	.	AMPSND EQU 460 ;(&) - AMPERSAND	
328	0027	.	.	QUOTE EQU 470 ;(') - SINGLE QUOTE	
329	0029	.	.	ARPARN EQU 510 ;(]) - RIGHT PARENTHESIS	
330	002B	.	.	PLUS EQU 530 ;PLUS SIGN	
331	002C	.	.	COMMA EQU 540 ;COMMA	
332	002D	.	.	MINUS EQU 550 ;MINUS SIGN	
333	002E	.	.	PERIOD EQU 560 ;(.) - PERIOD	
334	002F	.	.	SLANT EQU 570 ;(/) - SLANT	
335	0030	.	.	ZERO EQU 600 ;ASCII ZERO	
336	0032	.	.	TWO EQU 620 ;ASCII TWO	
337	0033	.	.	THREE EQU 630 ;ASCII THREE	
338	0034	.	.	FOUR EQU 640 ;ASCII FOUR	
339	0035	.	.	FIVE EQU 650 ;ASCII FIVE	
340	0036	.	.	SIX EQU 660 ;ASCII SIX	
341	0037	.	.	SEVEN EQU 670 ;ASCII SEVEN	
342	0000	.	.	;	
343	0040	.	.	ATSIGN EQU 1000 ;"AT" SIGN (@)	
344	0041	.	.	A EQU 1010 ;UPPER CASE A	
345	0043	.	.	C EQU 1030 ;UPPER CASE C	
346	0044	.	.	D EQU 1040 ;UPPER CASE D	
347	0046	.	.	F EQU 1060 ;UPPER CASE F	
348	0048	.	.	H EQU 1100 ;UPPER CASE H	
349	004C	.	.	L EQU 1140 ;UPPER CASE L	
350	004E	.	.	N EQU 1160 ;UPPER CASE N	
351	0052	.	.	R EQU 1220 ;UPPER CASE R	
352	0053	.	.	S EQU 1230 ;UPPER CASE S	
353	0054	.	.	T EQU 1240 ;UPPER CASE T	
354	0055	.	.	U EQU 1250 ;UPPER CASE U	
355	0059	.	.	Y EQU 1310 ;UPPER CASE Y	
356	005A	.	.	Z EQU 1320 ;UPPER CASE Z	
357	005B	.	.	LFTBKT EQU 1330 ;LEFT BRACKET	
358	005C	.	.	ABCKSL EQU 1340 ;(\) - BACK SLANT	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 10
=====
360      0000      . . .      ;*****
361      0000      . . .      ; LOWER CASE EQUATES *
362      0000      . . .      ;*****
363      0061      . . .      SMALLA EQU 141Q      ;LOWER CASE A
364      0063      . . .      ALCC EQU 143Q      ;ASCII LOWER CASE C
365      0064      . . .      SMALLD EQU 144Q      ;LOWER CASE D
366      0066      . . .      SMALLF EQU 146Q      ;LOWER CASE F
367      0069      . . .      SMALLI EQU 151Q      ;LOWER CASE I
368      006B      . . .      SMALLK EQU 153Q      ;LOWER CASE K
369      0070      . . .      SMALLP EQU 160Q      ;LOWER CASE P
370      0078      . . .      SMALLX EQU 170Q      ;LOWER CASE X
371      007B      . . .      LFTBRC EQU 173Q      ;LEFT BRACE
372      007C      . . .      VRTBAR EQU 174Q      ;VERTICAL BAR
373      007F      . . .      ADEL EQU 177Q      ;DELETE (RUBOUT)
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 11
=====
375      0000      . . .      ;*****
376      0000      . . .      ; DISPLAY FLAGS EQUATES *
377      0000      . . .      ;*****
378      00BF      . . .      ENHLIM EQU 277Q      ;MAXIMUM ENHANCEMENT CODE
379      00C0      . . .      STPR EQU 300Q      ;START PROTECTED FIELD
380      00C1      . . .      ENDPR EQU 301Q      ;END PROTECTED FIELD
381      00C2      . . .      XMONLY EQU 302Q      ;START TRANSMIT-ONLY FIELD
382      00C3      . . .      FILL EQU 303Q      ;EOL FILL CHARACTER
383      00C4      . . .      STPFLG EQU 304Q      ;NON-DISPLAYING TERMINATOR
384      00C5      . . .      ALPHA EQU 305Q      ;ALPHABETIC ONLY
385      00C6      . . .      NUMBER EQU 306Q      ;NUMERIC ONLY
386      00C7      . . .      ALPHNM EQU 307Q      ;ALPHANUMERIC FIELD
387      00C8      . . .      SFKYAT EQU 310Q      ;SOFT KEY ATTRIBUTE FIELD
388      0000      . . .      ;
389      00C4      . . .      FLDSEP EQU 304Q      ;FIELD SEPARATOR FOR I/O BUF
390      00CC      . . .      EOL EQU 314Q
391      00CE      . . .      EOP EQU 316Q
392      00D0      . . .      LNKLIM EQU 320Q      ;LOWEST VALUE FOR A LINK
393      0800      . . .      NUM2K EQU 4000Q      ;NUMBER 2048 (2K)
394      8000      . . .      B15 EQU 10000Q      ;BIT 15
395      00C3      . . .      JMP EQU 303Q      ;JUMP INSTRUCTION CODE
396      00C9      . . .      RET EQU 311Q      ;RETURN INSTRUCTION CODE
397      0000      . . .      ;*****
398      0000      . . .      ; MISCELLANEOUS EQUATES *
399      0000      . . .      ;*****
400      0017      . . .      MAXROW EQU 23      ;MAXIMUM ROW NUMBER
401      004F      . . .      MAXCOL EQU 79      ;MAXIMUM COLUMN NUMBER
402      0010      . . .      SFTEND EQU 16      ;LAST SOFT KEY DEFINITION RO
403      0008      . . .      BELLIM EQU 8      ;SPACE FROM RHTMGN FOR BELL
404      000F      . . .      BLKSM EQU 17Q      ;BLOCK SIZE MASK
405      0010      . . .      BLKSZ EQU 16      ;BLOCK SIZE
406      0001      . . .      REXMIT EQU 1Q      ;RE-TRANSMIT I/O FLAG
407      0002      . . .      BINXMT EQU 2      ;SEND BINARY DATA
408      0032      . . .      SFTDLY EQU 50      ;SOFT RESET PERIOD - .50 SEC
409      0080      . . .      NOSIGN EQU 200Q      ;NO SIGN FLAG FOR INPUT DATA
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 12
=====
411      0000      . . .      ;*****
412      0000      . . .      ; I/O MODULE EQUATES *
413      0000      . . .      ;*****
414      0000      . . .      RESET EQU 00          ;RESET TERMINAL VECTOR
415      0001      . . .      RSTJMP EQU 10         ;VECTOR FOR RESTART "PCHL"
416      0070      . . .      PROCSR EQU 1600       ;PROCESSOR "OUT" PORT
417      0080      . . .      IOBASE EQU 200Q      ;I/O ADDRESS MSB'S
418      0000      . . .      ;
419      0000      . . .      ;  KEYBOARD
420      0000      . . .      ;
421      8300      . . .      IOKB EQU 3Q+IOBASE*256 ;MODULE 11 BASE ADDRESS
422      8380      . . .      IOKBCO EQU IOKB+200Q ;RESET KEY CONTROL
423      0002      . . .      RSTON EQU 2Q          ;RESET ON
424      0004      . . .      RSTOFF EQU 4Q         ;RESET OFF
425      0008      . . .      NMFCTK EQU 8          ;NUMBER OF FUNCTION KEYS
426      0000      . . .      ;
427      0000      . . .      ;  CURSOR CONTROL
428      0000      . . .      ;
429      8700      . . .      IODISP EQU 7Q+IOBASE*256 ;MODULE 13 BASE ADDRESS
430      8700      . . .      IOCRCL EQU IODISP+0   ;CURSOR COLUMN ADDRESS
431      8720      . . .      IOCRRW EQU IODISP+40Q ;CURSOR ROW ADDRESS
432      0020      . . .      MAYEOP EQU 40Q        ;DMA ON, EOP IF DMA ROW = RO
433      0040      . . .      MAYEOL EQU 100Q       ;DMA OFF, SKIP EOP IF ROWS =
434      0060      . . .      DMAGFF EQU 140Q       ;DMA OFF
435      0080      . . .      CRTOFF EQU 200Q       ;DISPLAY OFF
436      0082      . . .      INVRS EQU 202Q        ;INVERSE VIDEO ON
437      0080      . . .      NORMAL EQU 200Q       ;NORMAL VIDEO ON
438      0000      . . .      ;
439      0000      . . .      ;  CARTRIDGE TAPE
440      0000      . . .      ;
441      8800      . . .      IOCTU EQU 13Q+IOBASE*256 ;MODULE 15 BASE ADDRES
442      8800      . . .      IOCTCO EQU IOCTU+0Q   ;COMMAND TO CTU
443      8800      . . .      IOCTSI EQU IOCTU+0Q   ;STATUS FROM CTU
444      8820      . . .      IOCTDO EQU IOCTU+40Q  ;DATA TO CTU
445      8820      . . .      IOCTDI EQU IOCTU+40Q  ;DATA FROM CTU
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
447	0000	.	.	;	13
448	0000	.	.	; 9866 PRINTER	
449	0000	.	.	;	
450	8000	.	.	IOPTR1 EQU 150+IOBASE*256 ;MODULE 16 BASE ADDRESS	
451	8020	.	.	PTROT1 EQU IOPTR1+400 ;PRINTER DATA OUT	
452	8000	.	.	PTRST1 EQU IOPTR1+00 ;PRINTER STATUS IN	
453	8002	.	.	PTRCL1 EQU IOPTR1+20 ;PRINTER CLEAR	
454	0000	.	.	;	
455	0000	.	.	; RS-232 PRINTER	
456	0000	.	.	;	
457	8500	.	.	IOPTR2 EQU 50+IOBASE*256 ;MODULE 12 BASE ADDRESS	
458	8540	.	.	PTROT2 EQU IOPTR2+1000 ;INTERFACE CONTROL OUT	
459	8520	.	.	PTRST2 EQU IOPTR2+400 ;PRINTER STATUS IN	
460	8560	.	.	PTRDA2 EQU IOPTR2+1400 ;PRINTER DATA OUT	
461	8540	.	.	PTRCF2 EQU IOPTR2+1000 ;OPTION JUMPERS IN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 14
=====
463      0000      . . .      ;*****
464      0000      . . .      ; PRINTER EQUATES *
465      0000      . . .      ;*****
466      0000      . . .      ;
467      0000      . . .      ; RS-232 OPTION STRAPS
468      0000      . . .      ;
469      0000      . . .      ;   BITS 2-0   MEANING IF SET
470      0000      . . .      ;   000       EXT BAUD RATE
471      0000      . . .      ;   001       110  "
472      0000      . . .      ;   010       150  "
473      0000      . . .      ;   011       300  "
474      0000      . . .      ;   100       1200 "
475      0000      . . .      ;   101       2400 "
476      0000      . . .      ;   110       4800 "
477      0000      . . .      ;   111       9600 "
478      0000      . . .      ;
479      0000      . . .      ;   BIT 3     PARITY SELECT
480      0000      . . .      ;   1         EVEN
481      0000      . . .      ;   0         ODD
482      0000      . . .      ;
483      0000      . . .      ;   BIT 4     PARITY INHIBIT
484      0000      . . .      ;   1         NO PARITY
485      0000      . . .      ;   0         PARITY
486      0000      . . .      ;   BITS 7-5  # OF FILLS
487      0000      . . .      ;   000       HANDSHAKE DEVICE
488      0000      . . .      ;   001       8
489      0000      . . .      ;   010       16
490      0000      . . .      ;   011       24
491      0000      . . .      ;   100       32
492      0000      . . .      ;   101       40
493      0000      . . .      ;   110       48
494      0000      . . .      ;   111       56
495      0000      . . .      ;*****
496      0000      . . .      ; DRIVER EQUATES *
497      0000      . . .      ;*****
498      05DC      . . .      PTDLY EQU 1500 ;15 SECOND PRINTER TIME OUT
499      0000      . . .      ;*****
500      0000      . . .      ; 9866 PRINTER EQUATES *
501      0000      . . .      ;*****
502      0001      . . .      PTRDY1 EQU 1 ;PRINTER READY
503      0080      . . .      PTRP01 EQU 200Q ;PRINTER OUT OF PAPER
504      0000      . . .      ;*****
505      0000      . . .      ; RS-232 PRINTER EQUATES *
506      0000      . . .      ;*****
507      0002      . . .      PTRDY2 EQU 2 ;PRINTER READY MASK
508      0040      . . .      PTRS82 EQU 100Q ;RS-232 SB LINE STROBE
509      0020      . . .      PTR0L2 EQU 40Q ;PRINTER READY MASK
510      00E0      . . .      PTRHD2 EQU 340Q ;RS-232 HANDSHAKE PROTOCOL
511      001F      . . .      PTRBD2 EQU 37Q ;PARITY AND BAUD RATE MASK
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
513	0000	. . .	;*****	15
514	0000	. . .	; VARIABLE SPACE ALLOCATION *	
515	0000	. . .	;*****	
516	FBFF	. . .	DSPLIM EQU 1757770 ;DISPLAY UPPER LIMIT	
517	00D0	. . .	LWDSP EQU 1500000/256 ;DISPLAY LOWER LIMIT	
518	FC00	. . .	IOBUF EQU 1760000	
519	00FC	. . .	IOBUFH EQU IOBUF/256	
520	0000	. . .	IOBUFL EQU -IOBUFH*256+IOBUF	
521	FC00	. . .	IOBUF1 EQU 1760000	
522	FD00	. . .	IOBUF2 EQU 1764000	
523	FE4F	. . .	DSPSTR EQU 1770000+79 ;MESSAGE BUFFER	
524	0100	. . .	PTRBLN EQU 256 ;PRINTER INPUT BUFFER SIZE	
525	0000	. . .	;*****	
526	0000	. . .	; OPERATING SYSTEM STORAGE *	
527	0000	. . .	;*****	
528	9160	. . .	STACK EQU FSTRAM+1400 ;STACK AREA (96 BYTES)	
529	FFD0	. . .	OPSTOR EQU 1777200 ;VARIABLES STORAGE AREA	
530	00FF	. . .	BASEH EQU OPSTOR/256 ;MSB OF DATA PAGE ADDRESSE	
531	FF00	. . .	BASE EQU BASEH*256 ;DATA PAGE BASE ADDRESS	
532	00FE	. . .	BASEH2 EQU BASEH-1 ;BASE VALUES FOR SECOND PAGE	
533	FE00	. . .	BASE2 EQU BASEH2*256 ;OF VARIABLES SPACE	
534	0000	. . .	;*****	
535	0000	. . .	; VARIABLE SUBROUTINE CALL *	
536	0000	. . .	;*****	
537	FFCD	. . .	ECONTF EQU OPSTOR-3 ;JUMP SUBROUTINE	
538	FFCE	. . .	CNTFAD EQU ECONSTF+1 ;CHARACTER FUNCTION ADDRESS	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 16
540	0000	. . .	;*****	
541	0000	. . .	; NORMAL/SOFT KEY SWAPPED DISPLAY PARAMETERS *	
542	0000	. . .	;*****	
543	FFCB	. . .	TOPLIN EQU ECONF-2 ;LSB PART OF NEXT LINE	
544	0000	. . .	; POINTER IN TOP DISPLAY	
545	0000	. . .	; LINE	
546	FFC9	. . .	LSTLIN EQU TOPLIN-2 ;POINTER TO LSB PART OF	
547	0000	. . .	; NEXT LINE POINTER IN	
548	0000	. . .	; LAST LINE PROCESSED	
549	FFC8	. . .	LSTCOL EQU LSTLIN-1 ;COLUMN AND ROW POSITION OF	
550	FFC7	. . .	LSTROW EQU LSTCOL-1 ;LAST CHARACTER PROCESSED	
551	0000	. . .	; (CORRESPONDS TO CHARACTER	
552	0000	. . .	; GIVEN BY "CURADR")	
553	FFC6	. . .	LSTDCD EQU LSTROW-1 ;LAST DISPLAY CODE USED	
554	FFC5	. . .	LSTFMT EQU LSTDCD-1 ;LAST FORMAT CONTROL USED	
555	FFC3	. . .	CURADR EQU LSTFMT-2 ;ADDRESS OF LAST CHARACTER	
556	0000	. . .	; PROCESSED	
557	FFC2	. . .	PROFLD EQU CURADR-1 ;PROTECT STATE OF (CURADR)	
558	0000	. . .	; = -1, PROTECTED	
559	0000	. . .	; # -1, NOT PROTECTED	
560	0000	. . .	;*****	
561	0000	. . .	; CURRENT CURSOR VALUES *	
562	0000	. . .	;*****	
563	FFC1	. . .	CURCOL EQU PROFLD-1 ;CURRENT COLUMN AND ROW	
564	FFC0	. . .	CURROW EQU CURCOL-1 ;POSITION OF CURSOR	
565	FFBF	. . .	LFTMGN EQU CURROW-1 ;LEFT MARGIN SETTING	
566	FFBE	. . .	RHTMGN EQU LFTMGN-1 ;RIGHT MARGIN SETTING	
567	000F	. . .	NUMSWP EQU ECONF-RHTMGN ;# OF SWAP VARIABLES	
568	FFAF	. . .	SWPSTR EQU RHTMGN-NUMSWP ;SWAP BUFFER	
569	FFAE	. . .	DSPTYP EQU SWPSTR-1 ;DISPLAY CURRENTLY ENABLED	
570	0000	. . .	; 0 = NORMAL DISPLAY	
571	0000	. . .	; -1 = SOFT KEY DISPLAY	
572	0000	. . .	;*****	
573	0000	. . .	; FIXED DISPLAY PARAMETERS (NOT SWAPPED) *	
574	0000	. . .	;*****	
575	FFAC	. . .	FRBLKS EQU DSPTYP-2 ;FREE BLOCKS LIST HEAD	
576	FFAA	. . .	DSPBGN EQU FRBLKS-2 ;LOW ADDRESS OF DISPLAY AREA	
577	FFA8	. . .	DSPEND EQU DSPBGN-2 ;HIGH ADDR OF DISPLAY AREA	
578	FFA6	. . .	SFTKYS EQU DSPEND-2 ;SOFT KEY DISPLAY START ADDR	
579	FFA4	. . .	CURFKY EQU SFTKYS-2 ;CURRENT FUNCTION KEY CHAR	
580	FFA3	. . .	TLINO EQU CURFKY-1 ;TOP LINE ABSOLUTE ROW NUMBE	
581	FFA1	. . .	LLINE EQU TLINO-2 ;LAST DISPLAY LINE START ADD	
582	FF9F	. . .	FLINE EQU LLINE-2 ;POINTER TO LSB PART OF NEXT	
583	0000	. . .	; LINE POINTER IN FIRST	
584	0000	. . .	; LINE OF NORMAL DISPLAY	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 17
586	0000	.	*****	
587	0000	.	; SCRATCH VARIABLES *	
588	0000	.	*****	
589	FF9E	.	TEMP1 EQU FLINE-1	
590	FF9D	.	TEMP EQU TEMP1-1 ;TEMPORARY STORAGE	
591	FF9C	.	CHARIN EQU TEMP-1 ;CHARACTER FROM KEYBOARD	
592	FF9B	.	NCHAR EQU CHARIN-1 ;NUMBER OF CHARS TO BE ADDED	
593	FF9A	.	NROWS EQU NCHAR-1 ;NO. OF ROWS TO BE ADDED	
594	FF99	.	NBLKS EQU NROWS-1 ;NO. OF BLOCKS TO BE ADDED	
595	FF98	.	CHSAV EQU NBLKS-1 ;SAVE AREA FOR CHAR	
596	0000	.	; PRECEDING LINK	
597	FF96	.	LNKSAV EQU CHSAV-2 ;LINK SAVE AREA	
598	FF94	.	EOLADR EQU LNKSAV-2 ;ADDR OF LAST EOL	
599	FF92	.	FRSTBL EQU EOLADR-2 ;FIRST BLUCK IN DISPL1	
600	FF91	.	BLKFIL EQU FRSTBL-1 ;FILL FLAG FOR FNDCHR	
601	FF90	.	EOLMV EQU BLKFIL-1 ;FLAG FOR EOLMOV	
602	FF8F	.	FILCHR EQU EOLMV-1 ;FILL CHAR SAVE FOR GTBLK	
603	CFFF	.	BFSPCE EQU 147777Q ;UPPER LIMIT OF BUFFER	
604	00B0	.	LWBUF EQU 130000Q/256 ;LOWER LIMIT	
605	FF8D	.	BUFBN EQU FILCHR-2 ;LOW ADDR OF NON-DISPLY BUFF	
606	FF8B	.	BUFEND EQU BUFBN-2 ;HIGH ADDR FOR BUFFER	
607	0000	.	*****	
608	0000	.	; STORAGE FOR CHARACTERS TO BE STORED *	
609	0000	.	*****	
610	FF8A	.	FMTCTL EQU BUFEND-1 ;FORMAT CONTROL TO BE ENTERE	
611	FF89	.	DCHAR EQU FMTCTL-1 ;NEXT CHAR TO BE DISPLAYED	
612	FF88	.	CHAR EQU DCHAR-1 ;CURRENT CHAR BEING PROCESSE	
613	FF86	.	CHKRTN EQU CHAR-2 ;CURRENT TYPE CHECK ROUTINE	
614	FF85	.	TMPCOL EQU CHKRTN-1 ;COLUMN # STORAGE FOR RCADDR	
615	0000	.	*****	
616	0000	.	; STORAGE FOR CURSOR POSITIONING *	
617	0000	.	*****	
618	FF84	.	COUNT EQU TMPCOL-1 ;NUMBER OF BYTES TO FILL	
619	FF83	.	NMROLL EQU COUNT-1 ;NUMBER OF LINES TO ROLL	
620	FF82	.	ROLLCT EQU NMROLL-1 ;ROLL COUNTER	
621	0000	.	;	
622	FFDB	.	NEWCOL EQU PARM1 ;NEW COLUMN NUMBER	
623	FFDA	.	NEWROW EQU PARM2 ;NEW ABSOLUTE ROW NUMBER	
624	FFD9	.	SCRNRW EQU PARM3 ;NEW SCREEN ROW SETTING	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 18
=====
626      0000      . . .      ;*****
627      0000      . . .      ; HORIZONTAL TAB TABLE *
628      0000      . . .      ;*****
629      000A      . . .      HTBLEN EQU 10 ;TABLE LENGTH (= 10 X 8)
630      FF78      . . .      HTBTBL EQU ROLLCT-HTBLEN
631      0000      . . .      ;*****
632      0000      . . .      ; DISPLAY SEND STORAGE *
633      0000      . . .      ;*****
634      FF77      . . .      CDSPEN EQU HTBTBL-1 ;CURRENT ENHANCEMENT IN
635      FF76      . . .      ENHOUT EQU CDSPEN-1 ;LAST ENHANCEMENT OUT
636      FF75      . . .      CALTST EQU ENHOUT-1 ;CURRENT ALTERNATE SET OUT
637      FF73      . . .      GETADR EQU CALTST-2 ;CURRENT CHARACTER ADDRESS
638      0000      . . .      ;*****
639      0000      . . .      ; FLAGS AND TABLE POINTERS *
640      0000      . . .      ;*****
641      FF72      . . .      CHRSET EQU GETADR-1 ;CURRENT ALTERNATE CHAR SET
642      FF71      . . .      KBFCTK EQU CHRSET-1 ;KEYBOARD FUNCTION CODE
643      0000      . . .      ;*****
644      FF70      . . .      MFLGS EQU KBFCTK-1 ;BLOCK TRANSFER PENDING FLAG
645      0000      . . .      ;*****
646      0100      . . .      SDC2 EQU 1Q*256 ;DC2 PENDING
647      0200      . . .      SSTAT EQU 2Q*256 ;TERMINAL STATUS PENDING
648      0400      . . .      SSTAT2 EQU 4Q*256 ;TERMINAL STATUS 2 PENDING
649      0800      . . .      SDVST EQU 10Q*256 ;DEVICE STATUS PENDING
650      1000      . . .      SCRSEN EQU 20Q*256 ;CURSOR SENSE PENDING
651      2000      . . .      SFCTKY EQU 40Q*256 ;FUNCTION KEY PENDING
652      4000      . . .      SENTER EQU 100Q*256 ;DISPLAY SEND PENDING
653      8000      . . .      SDVDUN EQU 200Q*256 ;DEVICE DONE PENDING
654      0000      . . .      ;*****
655      FF6F      . . .      MFLGS2 EQU MFLGS-1 ;MAIN CODE MODE FLAGS
656      0000      . . .      ;*****
657      0001      . . .      SDVREC EQU 1Q ;DEVICE RECORD PENDING
658      0002      . . .      SBINRY EQU 2Q ;BINARY RECORD PENDING
659      0004      . . .      RELSNS EQU 4Q ;RELATIVE CURSOR SENSE
660      0008      . . .      ESCINP EQU 10Q ;ESC RECEIVED IN BLOCK MODE
661      0010      . . .      FRSOUT EQU 20Q ;FIRST SOFT KEY DATA OUT
662      0020      . . .      WRPDEL EQU 40Q ;DELETE CHAR W/ WRAP AROUND
663      0040      . . .      WRPFLG EQU 100Q ;LINE WRAP AROUND OCCURRED
664      0080      . . .      NWRWST EQU 200Q ;NEW ABSOLUTE ROW SET
665      0000      . . .      ;*****
666      FF6E      . . .      DFLGS EQU MFLGS2-1 ;DATA TRANSFER FLAGS
667      0000      . . .      ;*****
668      0001      . . .      SDACOM EQU 001Q ;DATACOM/KEYBOARD
669      0002      . . .      CNTXFR EQU 2Q ;CONTINUE BUFFER TO DATA COM
670      0004      . . .      NOSEND EQU 4Q ;NO DISPLAY DATA TO SEND
671      0010      . . .      FCTK2D EQU 20Q ;FUNCTION KEY TO DISPLAY
672      0040      . . .      KBDLOK EQU 100Q ;KB LOCKED BY ESCAPE SEQUENC
673      0080      . . .      XBF2DS EQU 200Q ;I/O BUFFER TO DISPLAY MODE
=====

```

```

=====
ITEM   LOC   OBJECT CODE  SOURCE STATEMENTS                                     PAGE 19
=====
675   0000   . . .      ;*****
676   FF6D   . . .      TRMFCT EQU  DFLGS-1  ;NON-DISPLAYING TERMINATOR
677   0000   . . .      ;*****
678   FFFF   . . .      STPXFR EQU  -1      ;TERMINATE TRANSFER
679   0000   . . .      DELTRM EQU  0      ;DELETE TERMINATOR
680   0001   . . .      IGNTRM EQU  1      ;IGNORE TERMINATOR
681   0000   . . .      ;*****
682   FF6C   . . .      SPOWL EQU  TRMFCT-1 ;SPACE OVERWRITE LATCH
683   0000   . . .      ;*****
684   0020   . . .      SPOWON EQU  40Q    ;SPOW LATCH ON
685   00FF   . . .      SPOWOF EQU  377Q   ;SPOW LATCH OFF
686   0000   . . .      ;*****
687   FF6B   . . .      MLKROW EQU  SPOWL-1 ;MEMORY LOCK ROW
688   FF6A   . . .      MLKFLG EQU  MLKROW-1 ;MEMORY LOCK FLAG
689   FF69   . . .      LCHAR EQU  MLKFLG-1 ;LAST CHARACTER PROCESSED
690   FF68   . . .      TCHAR EQU  LCHAR-1 ;CURRENT TEST PATTERN CHAR
691   FF67   . . .      CRAFLG EQU  TCHAR-1 ;CURSOR ADVANCE FLAG
692   0000   . . .      ;*****
693   0000   . . .      ; POINTERS FOR BINARY LOADER *
694   0000   . . .      ;*****
695   FFD5   . . .      LADDR EQU  PARM6   ;BYTE ADDRESS PARAMETER
696   FFDE   . . .      LDATA EQU  IODATA  ;INPUT DATA ACCUMULATOR
697   FFD7   . . .      LCHKSM EQU  PARM5   ;16-BIT CHECKSUM
=====
  
```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 21
743	0000	. . .	;*****	
744	FF63	. . .	UNIT0 EQU IOFLG2-1 ;UNIT STATUS	
745	0000	. . .	;*****	
746	0001	. . .	LPM EQU 1Q ;TAPE AT OR BEFORE LOAD POIN	
747	0002	. . .	LSTFWD EQU 2Q ;TAPE LAST MOVED FORWARD	
748	0004	. . .	FPS EQU 4Q ;TAPE WRITE PROTECTED	
749	0008	. . .	CMDEXC EQU 10Q ;SUCCESSFUL COMMAND EXECUTIO	
750	0010	. . .	DBLHOL EQU 20Q ;DOUBLE HOLE FOUND	
751	0020	. . .	BOT EQU 40Q ;TAPE PAST BOT HOLES	
752	0040	. . .	LP EQU 100Q ;TAPE PAST LP HOLE	
753	0080	. . .	EW EQU 200Q ;TAPE PAST EW HOLE	
754	0000	. . .	;*****	
755	FF62	. . .	CNTRLO EQU UNIT0-1 ;DATA TRANSFER FLAGS: *	
756	0000	. . .	;*****	
757	0001	. . .	EOF EQU 1Q ;END OF FILE	
758	0002	. . .	EVD EQU 2Q ;END OF VALID DATA	
759	0004	. . .	HRDERR EQU 4Q ;HARD ERROR	
760	0008	. . .	SFTERR EQU 10Q ;SOFT ERROR	
761	0010	. . .	HRDER1 EQU 20Q ;INTERRUPT ERROR FLAG	
762	0020	. . .	WRTERR EQU 40Q ;WRITE ERROR	
763	0040	. . .	DATATR EQU 100Q ;DATA RECORDED	
764	0080	. . .	EOFINH EQU 200Q ;INHIBIT REPORTING EOF	
765	0000	. . .	;*****	
766	FF61	. . .	RELTAK EQU CNTRLO-1 ;GAP LENGTH COUNTER	
767	FF5F	. . .	ABSTAK EQU RELTAK-2 ;ABSOLUTE TACH COUNTER	
768	41B0	. . .	STRTAK EQU 40660Q ;STARTING VALUE FOR COUNTER	
769	FF5E	. . .	FILNUM EQU ABSTAK-1 ;CURRENT FILE NUMBER	
770	FF5D	. . .	SFTCNT EQU FILNUM-1 ;SOFT ERRORS PER PASS	
771	FF56	. . .	OTHER EQU SFTCNT-7 ;STORAGE FOR UNIT NOT SEL.	
772	0000	. . .	;*****	
773	FF55	. . .	CMND EQU OTHER-1 ;CURRENT CTU COMMAND: *	
774	0000	. . .	;*****	
775	0001	. . .	RUN EQU 1Q ;MOVE TAPE	
776	0002	. . .	FWD EQU 2Q ;FORWARD	
777	0004	. . .	FST EQU 4Q ;FAST	
778	0008	. . .	REC EQU 10Q ;RECORD	
779	0010	. . .	USL EQU 20Q ;SELECT LEFT UNIT	
780	0020	. . .	GEN EQU 40Q ;GAP GENERATE	
781	0040	. . .	ANR EQU 100Q ;LIGHT FOR RIGHT UNIT	
782	0080	. . .	ANL EQU 200Q ;LIGHT FOR LEFT UNIT	
783	0000	. . .	;*****	
784	0000	. . .	; INPDEV, OUTDEV, BXSTAT - I/O DEVICES *	
785	0000	. . .	;*****	
786	0001	. . .	LFTCTU EQU 1Q ;LEFT CARTRIDGE TAPE UNIT	
787	0002	. . .	RGTCTU EQU 2Q ;RIGHT CARTRIDGE TAPE UNIT	
788	0004	. . .	DISPLY EQU 4Q ;DISPLAY	
789	0008	. . .	PRINTR EQU 10Q ;PRINTER	
790	0010	. . .	ALTIO EQU 20Q ;ALTERNATE I/O	
791	0020	. . .	DATCOM EQU 40Q ;DATA COMM	
792	0040	. . .	PTTPOK EQU 100Q ;BUF AVAIL FOR PTTPLN ROUTIN	

13255
2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 22
=====
793      0080      . . .      BUFBSY EQU 200Q      ;BUF HELD BY UNSPECIFIED DEV
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 23
795	FF54	. . .	SCNCNT EQU CMND-1 ;NUM. OF KBSCAN PER CTU SCAN	
796	FF53	. . .	CTBLNK EQU SCNCNT-1 ;BLINK MASK FOR EJECT LIGHTS	
797	FF52	. . .	CTBLTM EQU CTBLNK-1 ;BLINK TIMER	
798	0020	. . .	CTBDLY EQU 40Q ;BLINK DELAY	
799	FF51	. . .	HOLCNT EQU CTBLTM-1 ;HOLE COUNTER	
800	FF50	. . .	TPSTAL EQU HOLCNT-1 ;TAPE STALL COUNTER	
801	0000	. . .	*****	
802	0000	. . .	; I/O VARIABLES *	
803	0000	. . .	*****	
804	FF4F	. . .	IOCERR EQU TPSTAL-1 ;I/O ERROR FLAG	
805	0000	. . .	; 0 = NO ERROR	
806	0000	. . .	; -1 = ERROR OCCURRED	
807	FF4E	. . .	INPDEV EQU IOCERR-1 ;CURRENT INPUT DEVICE	
808	FF4D	. . .	OUTDEV EQU INPDEV-1 ;CURRENT OUTPUT DEVICE	
809	FF4C	. . .	IOCPT EQU OUTDEV-1 ;DEVICE FLAG POINTER	
810	FF4B	. . .	IOSTA3 EQU IOCPT-1 ;DEVICE STATUS BYTE 3	
811	FF4A	. . .	IOSTA2 EQU IOSTA3-1 ;DEVICE STATUS BYTE 2	
812	FF49	. . .	IOSTA1 EQU IOSTA2-1 ;DEVICE STATUS BYTE 1	
813	FF48	. . .	IOSTA0 EQU IOSTA1-1 ;DEVICE NUMBER FOR STATUS	
814	FF47	. . .	XFRLIM EQU IOSTA0-1 ;TRANSFER LIMIT	
815	FF46	. . .	CMPLIM EQU XFRLIM-1 ;COMPARE LIMIT	
816	FF3D	. . .	B2DBUF EQU CMPLIM-9 ;BIN TO DECIMAL CONV BUFFER	
817	003D	. . .	B2DBFL EQU B2DBUF-BASE ;LSB PART OF "B2DBUF"	
818	FF3C	. . .	B2DPTR EQU B2DBUF-1 ;B2DBUF "GET" POINTER (LSB)	
819	FF3B	. . .	B2DEND EQU B2DPTR-1 ;B2DBUF END POINTER	
820	0000	. . .	; I/O CONTROL VARIABLES	
821	0000	. . .	; *****	
822	0000	. . .	; *****	
823	FFDB	. . .	IOCDEV EQU PARM1 ;DEVICE FLAG	
824	FFDA	. . .	IOCOUT EQU PARM2 ;OUTPUT DEVICE ACCUMULATOR	
825	FFD9	. . .	IOCINP EQU PARM3 ;INPUT DEVICE ACCUMULATOR	
826	FFD8	. . .	IOCTYP EQU PARM4 ;COMMAND MODIFIER FLAG	
827	FFD7	. . .	IOCMND EQU PARM5 ;COMMAND TYPE FLAG	
828	FFD5	. . .	IOCCNT EQU PARM6 ;DATA COUNT (2 BYTES)	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 24
=====
  830      0000      . . .      ;
  831      0000      . . .      ; I/O BUFFER INFORMATION STORAGE
  832      0000      . . .      ;
  833      FF3A      . . .      B1STAT EQU  B2DEND-1 ;STATUS OF FIRST BUFFER
  834      FF39      . . .      B1TYPE EQU  B1STAT-1 ;TYPE (-1=NORM, 0=EOF, 1=EVD)
  835      FF38      . . .      B1LEN  EQU  B1TYPE-1 ;LENGTH OF RECORD
  836      FF37      . . .      B2STAT EQU  B1LEN-1  ;STATUS OF SECOND BUFFER
  837      FF36      . . .      B2TYPE EQU  B2STAT-1 ;TYPE (-1=NORM, 0=EOF, 1=EVD)
  838      FF35      . . .      B2LEN  EQU  B2TYPE-1 ;LENGTH OF RECORD
  839      0000      . . .      ;
  840      0000      . . .      ; STORAGE FOR CARTRIDGE TAPE INTERRUPT ROUTINES
  841      0000      . . .      ;
  842      FF33      . . .      CTIADR EQU  B2LEN-2  ;ADDRESS (HAS SEVERAL USES)
  843      FF31      . . .      CTISPT EQU  CTIADR-2 ;POINTER TO BUFFER STATUS
  844      FF2F      . . .      CTIBPT EQU  CTISPT-2 ;POINTER TO BUFFER
  845      FF2C      . . .      CTICNT EQU  CTIBPT-3 ;GENERAL COUNTERS
  846      FF2B      . . .      CTITRL EQU  CTICNT-1 ;RE-READ COUNTER, HOLE CNTR
  847      FF2A      . . .      CTICSM EQU  CTITRL-1 ;CHECKSUM COUNTER
  848      FF29      . . .      CTISTA EQU  CTICSM-1 ;COMMAND SOURCE FLAG
  849      0000      . . .      ;
  850      0000      . . .      ; STORAGE FOR READ AND RECORD
  851      0000      . . .      ;
  852      FF27      . . .      NXTRED EQU  CTISTA-2 ;PTR INTO BUF FOR NEXT READ
  853      FF25      . . .      LSTRED EQU  NXTRED-2 ;PTR INTO BUF FOR READ REPEA
  854      FF24      . . .      SWPCTU EQU  LSTRED-1 ;SWAP CTU IN LOGGING MODE
  855      0000      . . .      ;
  856      0000      . . .      ;
  857      FF23      . . .      SAVINP EQU  SWPCTU-1 ;"INPDEV" SAVE FOR LOCAL RCR
  858      FF22      . . .      SAVOUT EQU  SAVINP-1 ;SAVE OUTDEV DURING LCL READ
  859      0000      . . .      ;
  860      0000      . . .      ; DATA FOR FORMAT DISPLAY STORAGE
  861      0000      . . .      ;
  862      FF21      . . .      ENDCOL EQU  SAVOUT-1 ;ENDING COLUMN AND ROW FOR
  863      FF20      . . .      ENDROW EQU  ENDCOL-1 ;PREV NON-PROTECTED FIELD
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 25
=====
865      0000      . . .      ;
866      0000      . . .      ; EXTENDED MAIN CODE RAM AREA
867      0000      . . .      ;
868      FE80      . . .      XTRASP EQU 177200Q
869      0000      . . .      ;*****
870      FE7F      . . .      DEVFLG EQU XTRASP-1 ;DEVICE PRESENT FLAG
871      0000      . . .      ;*****
872      0080      . . .      CTUIN EQU 200Q      ;CTU CODE PRESENT
873      0040      . . .      ALTIN EQU 100Q     ;ALTERNATE I/O PRESENT
874      0000      . . .      ;*****
875      0000      . . .      ; PRINTER VARIABLES *
876      0000      . . .      ;*****
877      0020      . . .      PTRINP EQU 40Q      ;ALLOW PRINTER INPUT
878      0000      . . .      ;*****
879      FE7D      . . .      PTRBBG EQU DEVFLG-2 ;START OF PRINTER BUFFER
880      FE7B      . . .      PTRSPT EQU PTRBBG-2 ;LOAD POINTER
881      FE79      . . .      PTRBPT EQU PTRSPT-2 ;UNLOAD POINTER
882      FE78      . . .      PTRABT EQU PTRBPT-1 ;PRINTER ERROR FLAG
883      0000      . . .      ; = 0, NO PRINTER ERROR
884      0000      . . .      ; = -1, PRINT ERROR OCCURRED
885      FE77      . . .      PTRFLG EQU PTRABT-1 ;PRINTER TYPE FLAG
886      0000      . . .      ; = 0, NO PRINTER
887      0000      . . .      ; = 1, PARALLEL INTERFACE
888      0000      . . .      ; = 2, RS-232 INTERFACE
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
 890      0000      . . .          ORG 24000Q
 891      2800      . . .          CTSTRT EQU $
 892      2800      54 . .         DB  VERSN      ;ROM PRESENT/VERSION FLAGS
 893      2801      28 . .         DB  CTSTRT/256 ;ADDRESS CHECK
 894      2802      . . .          ;
 895      2802      . . .          ; VECTORS TO I/O ROUTINES FOR MAIN CODE
 896      2802      . . .          ;
 897      2802      C3 5A 36         JMP GRNKEY
 898      2805      C3 9F 37         JMP REDKEY
 899      2808      C3 84 37         JMP CTLRED
 900      2808      C3 EF 37         JMP RECKEY
 901      280E      C3 04 37         JMP SELKEY
 902      2811      C3 E4 36         JMP CTUTST
 903      2814      C3 4C 36         JMP CONDTN
 904      2817      C3 18 29         JMP RSTCTU
 905      281A      C3 81 3D         JMP IOCNTL
 906      281D      C3 E1 45         JMP IOSTGQ
 907      2820      C3 15 46         JMP IODNGO
 908      2823      C3 8A 41         JMP IORDGO
 909      2826      C3 3B 38         JMP RCRDGO
 910      2829      C3 0D 43         JMP BNRVGO
 911      282C      C3 AF 3C         JMP CTDCDP
 912      282F      C3 6F 29         JMP CTMON
 913      2832      C3 C0 39         JMP PTTPLN
 914      2835      7D 28 .         DW  TID00
 915      2837      C3 75 43         JMP RDABRT
 916      283A      C3 C7 2D         JMP BSYCHK
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 27
=====
  918     283D     . . .      ;
  919     283D     . . .      ; * * * * *
  920     283D     . . .      ;
  921     283D     . . .      ;          CTU INTERRUPT - CALLED AFTER POLLING IN
  922     283D     . . .      ;          MAIN CODE
  923     283D     . . .      ;
  924     283D     . . .      ;          ENTRY:  INTERRUPT - PSW,H PUSHED
  925     283D     . . .      ;
  926     283D     . . .      ;          EXIT :  RETURN FROM INTERRUPT
  927     283D     . . .      ;          BITS FOR TAPES REMOVED SINCE LAST
  928     283D     . . .      ;          CALL TO CTMON REMAIN OFF.  IF TACH
  929     283D     . . .      ;          INTERRUPT OCCURRED, ABSTAK IS DECD.
  930     283D     . . .      ;
  931     283D     . . .      ;
  932     283D     . . .      ;          CTINTR EQU $
  933     283D     3A 66 FF      LDA CTSTAT      ;GET OLD STATUS
  934     2840     F6 FC .      ORI -1-CIL-CIR ;BITS CIL & CIR ARE ANDED,
  935     2842     21 00 8B      LXI H,IOCTSI   ;OTHERS TAKEN FROM NEW
  936     2845     A6 . . .      ANA M          ;STATUS WORD.
  937     2846     21 66 FF      LXI H,CTSTAT
  938     2849     6E . . .      MOV L,M        ;GET OLD STATUS
  939     284A     32 66 FF      STA CTSTAT     ;SAVE NEW STATUS
  940     284D     AD . . .      XRA L          ;HAS HOL STATUS CHANGED?
  941     284E     E6 10 .      ANI HOL
  942     2850     C4 89 28      CNZ HOLCHK     ;YES - DECIDE WHAT TO DO
  943     2853     2E 50 .      MVI L,TPSTAL  ;RESET TAPE STALL COUNTER
  944     2855     36 06 .      MVI M,6
  945     2857     2E 66 .      MVI L,CTSTAT  ;TACH INTERRUPT?
  946     2859     7E . . .      MOV A,M
  947     285A     B7 . . .      ORA A
  948     285B     F2 73 28      JP CTI100      ;NO -
  949     285E     2E 51 .      MVI L,HOLCNT  ;YES - DECR HOLE CHECK COUNT
  950     2860     35 . . .      DCR M
  951     2861     F2 66 28      JP CTI020     ;COUNT OVERFLOWED?
  952     2864     36 00 .      MVI M,0       ;YES - RESET TO 0
  953     2866     . . .      CTI020 EQU $   ;COUNT = 0?
  954     2866     CC D8 28      CZ HOLCT0     ;YES - RUNOFF OR LP HOLE
  955     2869     2E 5F .      MVI L,ABSTAK ;INC ABSOLUTE TACH COUNT
  956     286B     34 . . .      INR M         ;CARY FROM LOW BYTE?
  957     286C     C2 71 28      JNZ CTI040    ;NO -
  958     286F     23 . . .      INX H        ;YES - INC HIGH BYTE
  959     2870     34 . . .      INR M
  960     2871     . . .      CTI040 EQU $
  961     2871     2E 66 .      MVI L,CTSTAT  ;ROUTINES WANT H,L -> CTSTAT
  962     2873     . . .      CTI100 EQU $
  963     2873     3A 63 FF      LDA UNIT0     ;ROUTINES WANT A = UNIT0
  964     2876     CD E0 FF      CALL CTIJMP   ;PERFORM INTERRUPT ROUTINE
  965     2879     E1 . . .      POP H        ;RESTORE REGISTERS
  966     287A     F1 . . .      POP PSW
  967     287B     FB . . .      EI           ;RETURN FROM INTERRUPT
=====

```

13255
2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS          PAGE  28
=====
  968     287C    C9      .      .              RET
```



```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 29
=====
 970      287D      . . .      ;
 971      287D      . . .      ; "DO NOTHING" CTU INTERRUPT HANDLER
 972      287D      . . .      ;
 973      287D      . . .      TID00 EQU $
 974      287D      3A 20 8B    LDA IOCTDI ;CLEAR POSSIBLE BYTE RDY INT
 975      2880      7E . .     MOV A,M    ;GET CTU STATUS
 976      2881      2F . .     CMA       ;ARE BOTH TACH INTERRUPT
 977      2882      E6 A0 .    ANI TKI+GAP ;AND GAP SET?
 978      2884      C0 . .     RNZ       ;NO - QUIT
 979      2885      2E 61 .    MVI L,RELTAK ;YES - COUNT THE TACH EDGE
 980      2887      34 . .     INR M
 981      2888      C9 . .     RET
=====
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 31
1033	28AA	.	.	;*****	
1034	28AA	3F	.	CMC ;YES - TURN OFF EW	
1035	28AB	1F	.	RAR	
1036	28AC	77	.	MOV M,A	
1037	28AD	CD	E7 2A	CALL CHKFWD ;WHICH DIRECTION?	
1038	28B0	C8	.	RZ ;REVERSE - RETURN	
1039	28B1	C3	F1 28	JMP CTHNG1 ;FWD - HANG TIL TAPE REMOVED	
1040	28B4	.	.	;*****	
1041	28B4	.	.	; BEFORE EARLY WARNING *	
1042	28B4	.	.	;*****	
1043	28B4	.	.	HCK400 EQU \$	
1044	28B4	E6	80 .	ANI LP+LP ;PAST LOAD POINT?	
1045	28B6	C4	E7 2A	CNZ CHKFWD ;YES - WHICH DIRECTION?	
1046	28B9	3E	80 .	MVI A,EW	
1047	28BB	C2	9B 28	JNZ STUNTO ;FWD - SET EW AND QUIT	
1048	28BE	.	.	;*****	
1049	28BE	.	.	; AT OR BEFORE LOAD POINT *	
1050	28BE	.	.	;*****	
1051	28BE	.	.	HCK500 EQU \$	
1052	28BE	7E	.	MOV A,M ;2ND OF DOUBLE HOLE?	
1053	28BF	E6	10 .	ANI DBLHOL	
1054	28C1	C2	C9 28	JNZ HCK600 ;YES - SET FLAGS	
1055	28C4	2E	51 .	MVI L,HOLCNT ;NO - SET COUNT TO CHECK FOR	
1056	28C6	36	0C .	MVI M,12 ;DOUBLE HOLE	
1057	28C8	C9	.	RET	
1058	28C9	.	.	;*****	
1059	28C9	.	.	; JUST PASSED DOUBLE HOLE *	
1060	28C9	.	.	;*****	
1061	28C9	.	.	HCK600 EQU \$	
1062	28C9	AE	.	XRA M ;CLEAR DOUBLE HOLE FLAG	
1063	28CA	F6	01 .	ORI LPM ;SET "AT OR BEFORE LP"	
1064	28CC	E6	9F .	ANI -1-LP-BOT ;SET "BEFORE BOT"	
1065	28CE	77	.	MOV M,A	
1066	28CF	CD	E7 2A	CALL CHKFWD ;WHICH DIRECTION?	
1067	28D2	C8	.	RZ ;REVERSE - RET "BEFORE BOT"	
1068	28D3	7E	.	MOV A,M ;FORWARD - RET "AFTER BOT"	
1069	28D4	F6	20 .	ORI BOT	
1070	28D6	77	.	MOV M,A	
1071	28D7	C9	.	RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 32
=====
1073     28D8      . . .      ;
1074     28D8      . . .      ; * * * * *
1075     28D8      . . .      ;
1076     28D8      . . .      ;      HOLCT0 - HOLE CHECK COUNT WENT TO 0
1077     28D8      . . .      ;
1078     28D8      . . .      ;      INTERPRETATION:  IN HOLE => RUNOFF
1079     28D8      . . .      ;                        NOT IN HOLE => SINGLE HOLE
1080     28D8      . . .      ;
1081     28D8      . . .      ;      CALLED ONLY BY CTU INTERRUPT
1082     28D8      . . .      ;
1083     28D8      . . .      ;      ENTRY:  A = CURRENT STATUS
1084     28D8      . . .      ;                H = BASEH
1085     28D8      . . .      ;
1086     28D8      . . .      ;      EXIT :  H = BASEH
1087     28D8      . . .      ;                A,L DESTROYED
1088     28D8      . . .      ;
1089     28D8      . . .      ;      HOLCT0 EQU $
1090     28D8      E6 10      .      ANI  HOL          ;IN HOLE?
1091     28DA      C2 EB 28      JNZ  CTHNGO       ;YES - HANG TIL TAPE REMOVED
1092     28DD      . . .      ;*****
1093     28DD      . . .      ; JUST PASSED LOAD POINT HOLE *
1094     28DD      . . .      ;*****
1095     28DD      3E 61      .      MVI  A,BOT+LP+LPM
1096     28DF      CD 9B 28      CALL STUNTO       ;SET FOR GOING FWD
1097     28E2      CD E7 2A      CALL CHKPWD       ;WHICH DIRECTION?
1098     28E5      C0 . .      RNZ                ;FWD - RETURN
1099     28E6      7E . .      MOV  A,M          ;REV -
1100     28E7      E6 BF .      ANI  -1-LP        ;MARK BEFORE LP
1101     28E9      77 . .      MOV  M,A
1102     28EA      C9 . .      RET
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
1104	28EB	.	.	;
1105	28EB	.	.	; * * * * *
1106	28EB	.	.	;
1107	28EB	.	.	; CTHANG - HANG UNTIL TAPE IS REMOVED
1108	28EB	.	.	;
1109	28EB	.	.	; ENTRY CTHNGO - RUNOFF
1110	28EB	.	.	;
1111	28EB	.	.	; ENTRY CTHNG1 - RAN INTO END-OF-TAPE HOLE
1112	28EB	.	.	;
1113	28EB	.	.	; EXIT : C,Z
1114	28EB	.	.	; H = BASEH
1115	28EB	.	.	; A,L DESTROYED
1116	28EB	.	.	;
1117	28EB	.	.	CTHNGO EQU \$
1118	28EB	21	F4 3A	LXI H,OFFMSG ;GET RUNOFF MESSAGE
1119	28EE	C3	F4 28	JMP CTHANG
1120	28F1	.	.	CTHNG1 EQU \$
1121	28F1	21	FC 3A	LXI H,UETMSG ;GET "UNEXPECTED END OF TP"
1122	28F4	.	.	CTHANG EQU \$
1123	28F4	E5	.	PUSH H ;SAVE MESSAGE POINTER
1124	28F5	CD	E4 28	CALL STOPTP ;STOP THE TAPE
1125	28F8	CD	14 48	CALL ZBELL ;RING THE BELL
1126	28FB	E1	.	POP H
1127	28FC	CD	A3 3C	CALL CARDIO ;DISPLAY MESSAGE
1128	28FF	CD	DC 2A	CALL GTCTBT ;GET BIT FOR STALLED UNIT
1129	2902	2F	.	CMA ;TURN IT OFF IN CTSTAT SO
1130	2903	21	66 FF	LXI H,CTSTAT ;SOFT RESET WILL CAUSE
1131	2906	A6	.	ANA M ;REWIND ATTEMPT
1132	2907	77	.	MOV M,A
1133	2908	.	.	CTH100 EQU \$
1134	2908	CD	C1 29	CALL CTMON1 ;CHECK OTHER TAPE
1135	2908	CD	DC 2A	CALL GTCTBT ;GET BIT FOR THIS UNIT
1136	290E	21	00 8B	LXI H,IOCTSI ;HAS IT BEEN REMOVED?
1137	2911	A6	.	ANA M
1138	2912	C2	08 29	JNZ CTH100 ;NO - CONTINUE WAITING
1139	2915	CD	43 00	CALL RSTDSP ;TAPE REMOVED - RESTORE DISP
1140	2918	26	FF	MVI H,BASEH
1141	291A	C9	.	RET

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 34
=====
1143     291B      . . . ;
1144     291B      . . . ; * * * * *
1145     291B      . . . ;
1146     291B      . . . ; RSTCTU - CTU SOFT RESET
1147     291B      . . . ;
1148     291B      . . . ; ENTRY: INTERRUPTS DISABLED
1149     291B      . . . ;
1150     291B      . . . ; EXIT : INTERRUPTS ENABLED
1151     291B      . . . ; ALL REGISTERS DESTROYED
1152     291B      . . . ;
1153     291B      . . . ;
1154     291B      . . . RSTCTU EQU $
1155     291B      21 00 00 LXI H,0 ;CLEAR TRANSFER FLAGS
1156     291E      22 64 FF SHLD IOFLG2
1157     2921      CD 17 3D CALL FREBFS ;FREE I/O BUFFERS
1158     2924      CD 97 3D CALL IOERCL ;CLEAR ERROR (IOCERR <- S)
1159     2927      3A 55 FF LDA CMND ;GET LAST-ISSUED COMMAND
1160     292A      F5 . . . PUSH PSW ;SAVE IT
1161     292B      CD E4 2B CALL STOPTP ;STOP TAPE, RESET LIGHTS, EI
1162     292E      F1 . . . POP PSW ;RECALL COMMAND
1163     292F      0F . . . RRC ;WAS TAPE RUNNING WHEN RESET
1164     2930      D0 . . . RNC ;NO - RETURN
1165     2931      07 . . . RLC ;YES - RESTORE COMMAND
1166     2932      E6 08 . ANI REC ;WAS TAPE RECORDING?
1167     2934      CA 69 29 JZ RCT120 ;NO - REWIND TO LP & EXIT
1168     2937      . . . ;*****
1169     2937      . . . ; CTU WAS RECORDING - TRY TO RECOVER *
1170     2937      . . . ;*****
1171     2937      . . . RCT020 EQU $
1172     2937      3A 63 FF LDA UNIT0 ;FINDING LOAD POINT?
1173     293A      E6 01 . ANI LPM
1174     293C      C2 69 29 JNZ RCT120 ;YES - JUST REWIND TO LOAD P
1175     293F      21 02 00 LXI H,2 ;BACKSPACE OVER A BAD RECORD
1176     2942      CD 79 2C CALL BAKSPR ;AND A GOOD ONE
1177     2945      D4 AD 2B CNC CHKLPM ;FWD TO LOAD PT IF NOT THERE
1178     2948      DA 84 36 JC USREXT ;REPORT ANY ERRORS
1179     294B      CA 64 29 JZ RCT100 ;AT LOAD POINT - DO NOT READ
1180     294E      3E BF . MVI A,-1-DATATR
1181     2950      CD A5 32 CALL CLRCT0 ;CLEAR "DATA RECORDED" FLAG
1182     2953      3E FF . MVI A,-1 ;SET XFR LIMIT TO ONE RECORD
1183     2955      32 47 FF STA XFRLIM
1184     2958      CD DC 2A CALL GTCTBT ;GET BIT FOR THIS UNIT
1185     295B      47 . . . MOV B,A
1186     295C      CD 12 2E CALL CT2BUF ;IF NOT AT LP, READ A RECORD
1187     295F      . . . (TO SET EOF CORRECTLY)
1188     295F      DA 84 36 JC USREXT ;REPORT ANY ERRORS
1189     2962      97 . . . SUB A ;NO ERROR - FREE BUFFER
1190     2963      12 . . . STAX D
1191     2964      . . . RCT100 EQU $
1192     2964      3E 40 . MVI A,DATATR ;SET "DATA RECORDED" FLAG
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 35
=====
1193     2966     CD  72  32          CALL SETCT0
1194     2969     .   .   .          RCT120 EQU $
1195     2969     CD  75  28          CALL RWDLP      ;REWIND TO LOAD POINT
1196     296C     C3  84  36          JMP  USREXT     ;REPORT ANY ERRORS
=====
```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 37
=====
1248     29A4     C2  A9  29                JNZ  CTM040      ;YES -
1249     29A7     3E  02  .                MVI  A,CIR      ;NO - RIGHT TAPE INSERTED
1250     29A9     .   .   .                CTM040 EQU $
1251     29A9     67  .   .                MOV  H,A        ;SAVE BIT
1252     29AA     85  .   .                JRA  L          ;SET BIT IN STATUS FOR NEW
1253     29AB     32  66  FF               STA  CTSTAT     ;TAPE
1254     29AE     7C  .   .                MOV  A,H        ;RECALL BIT
1255     29AF     CD  89  2D               CALL SELACT     ;SELECT TAPE INDICATED
1256     29B2     CD  97  3D               CALL IOERCL     ;CLEAR ERR FLG (IOCERR <- S)
1257     29B5     21  00  21               LXI  H,BOT+LPM*256 ;SET UNIT0 AND CNTRL0
1258     29B8     22  62  FF               SHLD CNTRL0
1259     29BB     CD  75  2B               CALL RWDLP      ;REWIND TO LOAD POINT
1260     29BE     C3  84  36               JMP  USREXT     ;REPORT ANY ERRORS
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
1262	29C1	. . .	;	
1263	29C1	. . .	; * * * * *	
1264	29C1	. . .	;	
1265	29C1	. . .	; CTMON1 - CHECK FOR REMOVED TAPES	
1266	29C1	. . .	;	
1267	29C1	. . .	; ENTRY: DON'T CARE	
1268	29C1	. . .	;	
1269	29C1	. . .	; EXIT : L = OLD STATUS	
1270	29C1	. . .	; C => ERROR, RUN ISSUED TO ABSENT TP	
1271	29C1	. . .	; NC => NO SUCH ERROR	
1272	29C1	. . .	; Z => TAPE NOT RUNNING	
1273	29C1	. . .	; NZ => TAPE RUNNING	
1274	29C1	. . .	; A,H DESTROYED	
1275	29C1	. . .	;	
1276	29C1	. . .	;	
1277	29C1	. . .	CTMON1 EQU \$	
1278	29C1	CD 46 2A	CALL GETSTA ;GET STATUS	
1279	29C4	6F . .	MOV L,A ;SAVE IN L-REG	
1280	29C5	E6 01 .	ANI CIL ;LEFT TAPE REMOVED?	
1281	29C7	CC 5E 2A	CZ LITOFL ;YES - TURN OFF LIGHT	
1282	29CA	7D . .	MOV A,L ;RECALL STATUS	
1283	29CB	E6 02 .	ANI CIR ;RIGHT TAPE REMOVED?	
1284	29CD	CC 63 2A	CZ LITOFR ;YES - TURN OFF LIGHT	
1285	29D0	F3 . .	DI	
1286	29D1	3A 55 FF	LDA CMND ;GET CURRENT COMMAND	
1287	29D4	E6 01 .	ANI RUN ;RUN COMMAND ISSUED?	
1288	29D6	3A 50 FF	LDA TPSTAL ;(GET CORRESPONDING TPSTAL	
1289	29D9	FB . .	EI	
1290	29DA	C8 . .	RZ ;NO - RETURN NC, Z	
1291	29DB	B7 . .	ORA A ;TPSTAL = 0 => NO INTERRUPTS	
1292	29DC	C0 . .	RNZ ;INTERRUPTS OK - RET NC, NZ	
1293	29DD	CD E4 2B	CALL STOPTP ;STOP THE MOTOR	
1294	29E0	CD ED 2A	CALL CISCAN ;TAPE INSERTED?	
1295	29E3	21 63 3C	LXI H,STALMS ;(GET STALL MESSAGE)	
1296	29E6	D4 F4 2B	CNC CTHANG ;YES - REPORT STALL	
1297	29E9	3A 55 FF	LDA CMND ;WAS UNIT RECORDING?	
1298	29EC	E6 08 .	ANI REC	
1299	29EE	CA 05 2A	JZ CTM100 ;NO - RETURN	
1300	29F1	3E 20 .	MVI A,WRTErr ;YES - SET WRITE ERROR FLAG	
1301	29F3	CD 72 32	CALL SETCT0	
1302	29F6	CD DC 2A	CALL GTCTBT ;CLEAR BITS FOR THIS	
1303	29F9	2F . .	CMA ;UNIT IN I/O BUFFERS	
1304	29FA	F5 . .	PUSH PSW	
1305	29FB	21 3A FF	LXI H,B1STAT	
1306	29FE	A6 . .	ANA M	
1307	29FF	77 . .	MOV M,A	
1308	2A00	2E 37 .	MVI L,82STAT*256/256	
1309	2A02	F1 . .	POP PSW	
1310	2A03	A6 . .	ANA M	
1311	2A04	77 . .	MOV M,A	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                PAGE 39
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1312	2A05	.	.	CTM100 EQU \$	
1313	2A05	3A	66 FF	LDA CTSTAT ;RETURN L = STATUS	
1314	2A08	6F	.	MOV L,A	
1315	2A09	97	.	SUB A	
1316	2A0A	37	.	STC ;RETURN C, Z	
1317	2A0B	C9	.	RET	

```
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 40
=====
1319     2A0C      . . .      ;
1320     2A0C      . . .      ; * * * * *
1321     2A0C      . . .      ;
1322     2A0C      . . .      ; LOGRWD - DO DATA LOGGING-MODE REWIND, I.E.,
1323     2A0C      . . .      ; RECORD EVERYTHING ON THE SCREEN FIRST,
1324     2A0C      . . .      ; AND MONITOR THE DATACOM BETWEEN
1325     2A0C      . . .      ; OPERATIONS.
1326     2A0C      . . .      ;
1327     2A0C      . . .      ; ENTRY: SWPCTU CONTAINS BIT FOR CTU TO BE
1328     2A0C      . . .      ; REWOUND.
1329     2A0C      . . .      ; OUTDEV HAS BEEN UPDATED.
1330     2A0C      . . .      ;
1331     2A0C      . . .      ; EXIT : SWPCTU = -1 (=> TAPE SWAP MODE)
1332     2A0C      . . .      ; A,H,L DESTROYED.
1333     2A0C      . . .      ;
1334     2A0C      . . .      ;
1335     2A0C      . . .      LOGRWD EQU $
1336     2A0C      C5 . . .      PUSH B ;SAVE REGISTERS
1337     2A0D      D5 . . .      PUSH D
1338     2A0E      CD 97 3D      CALL IOERCL ;CLEAR ERR FLG (IOCERR <- S)
1339     2A11      CD 56 3A      CALL USREDA ;GET EVERYTHING OFF SCREEN
1340     2A14      3E 05 .      MVI A,5 ;WRITE EOF ON OLD UNIT
1341     2A16      CD 2D 2A      CALL LOG900
1342     2A19      3E 06 .      MVI A,6 ;WRITE EVD ON OLD UNIT
1343     2A1B      D4 2D 2A      CNC LOG900
1344     2A1E      3E 00 .      MVI A,0 ;REWIND OLD UNIT
1345     2A20      D4 2D 2A      CNC LOG900
1346     2A23      3E FF .      MVI A,-1 ;RESTORE SWPCTU
1347     2A25      32 24 FF      STA SWPCTU
1348     2A28      D1 . . .      POP D ;RESTORE REGISTERS
1349     2A29      C1 . . .      POP B
1350     2A2A      C3 84 36      JMP USREXT ;REPORT ANY ERRORS
1351     2A2D      . . .      ;*****
1352     2A2D      . . .      ; SUBROUTINE PERFORMS CONTROL FUNCTIONS ON *
1353     2A2D      . . .      ; TAPE JUST FINISHED *
1354     2A2D      . . .      ;*****
1355     2A2D      . . .      LOG900 EQU $
1356     2A2D      32 D8 FF      STA IOCTYP ;SAVE FUNCTION FOR CTLCT
1357     2A30      . . .      LOG910 EQU $
1358     2A30      CD 94 00      CALL GETDCM ;MONITOR DATACOM
1359     2A33      CA 30 2A      JZ LOG910 ;CHARS INPUT - KEEP MONITRNG
1360     2A36      CD C1 29      CALL CTMON1 ;CTU BUSY?
1361     2A39      C2 30 2A      JNZ LOG910 ;YES - WAIT
1362     2A3C      D8 . . .      RC ;(RETURN IF ANY ERRORS)
1363     2A3D      3A 24 FF      LDA SWPCTU ;NO - GET UNIT
1364     2A40      21 4D 41      LXI H,CTLTAB
1365     2A43      C3 94 41      JMP SETJMP ;PERFORM FUNCTION
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1367	2A46	.	.	;	41
1368	2A46	.	.	; * * * * *	
1369	2A46	.	.	;	
1370	2A46	.	.	; GETSTA - GET CTU STATUS	
1371	2A46	.	.	;	
1372	2A46	.	.	; GET AND SAVE CTU STATUS, CLEARING	
1373	2A46	.	.	; "CART. INSERTED" BITS FOR TAPES THAT HAVE	
1374	2A46	.	.	; BEEN REMOVED.	
1375	2A46	.	.	;	
1376	2A46	.	.	; EXIT : CTU NOT BUSY, A = NEW STATUS	
1377	2A46	.	.	; CTU RUNNING, A = OLD STATUS	
1378	2A46	.	.	;	
1379	2A46	.	.	;	
1380	2A46	.	.	GETSTA EQU \$	
1381	2A46	3A	55 FF	LDA CMND ;TAPE BUSY?	
1382	2A49	E6	01 .	ANI RUN	
1383	2A48	21	66 FF	LXI H,CTSTAT ;(GET OLD STATUS)	
1384	2A4E	7E	. .	MOV A,M	
1385	2A4F	C0	. .	RNZ ;YES - RETURN	
1386	2A50	3A	00 8B	LDA IOCTSI ;NO - GET NEW STATUS	
1387	2A53	F6	FC .	ORI -1-CIL-CIR ;"AND" BITS FOR TAPES,	
1388	2A55	A6	. .	ANA M ;OTHERS FRM OLD STATUS	
1389	2A56	77	. .	MOV M,A ;SAVE STATUS	
1390	2A57	C9	. .	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1392	2A58	.	.	;	42
1393	2A58	.	.	; * * * * *	
1394	2A58	.	.	;	
1395	2A58	.	.	; LITOFF - TURN OFF LIGHT ON SELECTED UNIT	
1396	2A58	.	.	;	
1397	2A58	.	.	; READS UNIT FROM CURRENT CONTENTS OF "CMND"	
1398	2A58	.	.	;	
1399	2A58	.	.	; EXIT : A DESTROYED	
1400	2A58	.	.	;	
1401	2A58	.	.	; LITOFFL - TURNS OFF LEFT LIGHT	
1402	2A58	.	.	; LITOFFR - TURNS OFF RIGHT LIGHT	
1403	2A58	.	.	;	
1404	2A58	.	.	;	
1405	2A58	.	.	LITOFF EQU \$	
1406	2A58	CD	DC 2A	CALL GTCTBT ;LEFT UNIT SELECTED?	
1407	2A5B	CA	63 2A	JZ LITOFFR ;NO - TURN OFF RIGHT LIGHT	
1408	2A5E	.	.	LITOFFL EQU \$	
1409	2A5E	3E	7F .	MVI A,-1-ANL ;SET UP MASK	
1410	2A60	C3	65 2A	JMP LOF010	
1411	2A63	.	.	LITOFFR EQU \$	
1412	2A63	3E	BF .	MVI A,-1-ANR ;MASK FOR RIGHT LIGHT	
1413	2A65	.	.	LOF010 EQU \$	
1414	2A65	E5	.	PUSH H ;SAVE H,L	
1415	2A66	F5	.	PUSH PSW ;SAVE MASK	
1416	2A67	21	53 FF	LXI H,CTBLNK ;TURN OFF BLINKING	
1417	2A6A	F3	.	DI ;HOLD OFF INTERRUPTS	
1418	2A6B	A6	.	ANA M	
1419	2A6C	77	.	MOV M,A	
1420	2A6D	F1	.	POP PSW ;RETRIEVE MASK	
1421	2A6E	2E	55 .	MVI L,CMND-BASE ;TURN OFF LIGHT IN "CMND"	
1422	2A70	A6	.	ANA M	
1423	2A71	77	.	MOV M,A	
1424	2A72	32	00 8B	STA IOCTCO ;OUTPUT COMMAND TO HARDWARE	
1425	2A75	FB	.	EI ;LITOFF IS ONLY CALLED WITH	
1426	2A76	.	.	INTERRUPTS DISABLED	
1427	2A76	E1	.	POP H ;RESTORE REGS	
1428	2A77	C9	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1430	2A78	.	.	;	43
1431	2A78	.	.	; * * * * *	
1432	2A78	.	.	;	
1433	2A78	.	.	; OUTCMD - ISSUE A COMMAND TO THE CTU	
1434	2A78	.	.	;	
1435	2A78	.	.	; OUTPUTS THE COMMAND AND SAVES IT IN CMND.	
1436	2A78	.	.	; UNIT SELECT IS PRESERVED FROM OLD COMMAND	
1437	2A78	.	.	; IF RUN BIT IS UNCHANGED, LIGHTS ARE UNCHNGD	
1438	2A78	.	.	; "RUN" TURNED ON - SELECTED LIGHT TURNED OFF	
1439	2A78	.	.	; BLINK TURNED ON.	
1440	2A78	.	.	; "RUN" TURNED OFF - LIGHT TURNED ON, BLINK	
1441	2A78	.	.	; TURNED OFF.	
1442	2A78	.	.	;	
1443	2A78	.	.	; ENTRY OCM001 DOES NOT CLEAR LPM (AT OR	
1444	2A78	.	.	; BEFORE LOAD POINT) OR SET CMDEXC (COMMAND	
1445	2A78	.	.	; EXECUTION OK).	
1446	2A78	.	.	;	
1447	2A78	.	.	; ENTRY: A = BITS RUN, FWD, FST, REC, GEN	
1448	2A78	.	.	; TO BE TURNED ON	
1449	2A78	.	.	; H,L -> INTERRUPT ROUTINE	
1450	2A78	.	.	;	
1451	2A78	.	.	; ENTRY OUTCM1 DOES NOT TAKE INTERRUPT	
1452	2A78	.	.	; ROUTINE PARAMETER	
1453	2A78	.	.	;	
1454	2A78	.	.	; EXIT : H = BASEH	
1455	2A78	.	.	; L DESTROYED	
1456	2A78	.	.	; NC (REQUIRED BY OTHER ROUTINES)	
1457	2A78	.	.	;	
1458	2A78	.	.	;	
1459	2A78	.	.	OUTCMD EQU \$	
1460	2A78	22	E1 FF	SHLD CTIVEC ;SET UP INTERRUPT VECTOR	
1461	2A78	.	.	OUTCM1 EQU \$	
1462	2A78	C5	.	PUSH B	
1463	2A7C	E6	2F .	ANI -1-USL-ANL-ANR ;TURN OFF BITS SET BY	
1464	2A7E	47	.	MOV B,A ;THIS ROUTINE - SAVE RESULT	
1465	2A7F	21	63 FF	LXI H,UNIT0	
1466	2A82	7E	.	MOV A,M ;GET UNIT FLAGS	
1467	2A83	E6	FE .	ANI -1-LPM ;ASSUME OPERATION WILL MOVE	
1468	2A85	.	.	; TAPE BEYOND LP	
1469	2A85	F6	08 .	ORI CMDEXC ;ASSUME OPERATION WILL RUN	
1470	2A87	77	.	MOV M,A	
1471	2A88	C3	92 2A	JMP OCM005	
1472	2A88	.	.	OCM000 EQU \$	
1473	2A88	22	E1 FF	SHLD CTIVEC	
1474	2A8E	.	.	OCM001 EQU \$;FAST OUTCMD ENTRY	
1475	2A8E	C5	.	PUSH B ;SAVE B,C	
1476	2A8F	E6	2F .	ANI -1-USL-ANL-ANR ;TURN OFF BITS SET BY	
1477	2A91	47	.	MOV B,A ;THIS ROUTINE - SAVE RESULT	
1478	2A92	.	.	OCM005 EQU \$	
1479	2A92	3A	55 FF	LDA CMND ;GET OLD COMMAND	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 44
1480	2A95	4F . .	MOV C,A ;SAVE IT	
1481	2A96	E6 D0 .	ANI USL+ANL+ANR ;USE OLD LIGHTS AND UNIT	
1482	2A98	80 . .	ORA B ;FORM NEW COMMAND	
1483	2A99	47 . .	MOV B,A ;SAVE NEW COMMAND	
1484	2A9A	A9 . .	XRA C ;SEE IF RUN BIT HAS CHANGED	
1485	2A9B	E6 01 .	ANI RUN	
1486	2A9D	78 . .	MOV A,B ;(RECALL COMMAND)	
1487	2A9E	CA BE 2A	JZ OCM030 ;NO - QUIT	
1488	2AA1	E6 10 .	ANI USL ;YES - GET BIT FOR LIGHT	
1489	2AA3	0E 80 .	MVI C,ANL ;OF SELECTED UNIT	
1490	2AA5	C2 AA 2A	JNZ OCM010	
1491	2AA8	0E 40 .	MVI C,ANR	
1492	2AAA	. . .	OCM010 EQU \$	
1493	2AAA	78 . .	MOV A,B ;RECALL COMMAND	
1494	2AAB	B1 . .	ORA C ;TURN ON SELECTED LIGHT	
1495	2AAC	47 . .	MOV B,A ;NOW CHECK RUN BIT	
1496	2AAD	E6 01 .	ANI RUN ;RUN TURNED ON OR OFF?	
1497	2AAF	C2 B4 2A	JNZ OCM020 ;ON - DO NOT CHANGE MASK	
1498	2AB2	0E 00 .	MVI C,0 ;OFF - CLEAR MASK	
1499	2AB4	. . .	OCM020 EQU \$	
1500	2AB4	3E 20 .	MVI A,CTBDLY ;RESET BLINK COUNTER	
1501	2AB6	32 52 FF	STA CTBLTM	
1502	2AB9	79 . .	MOV A,C ;GET MASK	
1503	2ABA	32 53 FF	STA CTBLNK ;0 => NO BLINK	
1504	2ABD	A8 . .	XRA B ;NO RUN - LIGHT LEFT ON,	
1505	2ABE	. . .	; RUN - LIGHT TURNED OFF	
1506	2ABE	. . .	OCM030 EQU \$;RETURN	
1507	2ABE	32 55 FF	STA CMND ;SAVE IN CMND	
1508	2AC1	32 00 8B	STA IOCTCO ;OUTPUT TO CTU	
1509	2AC4	21 50 FF	LXI H,TPSTAL ;RESET TAPE STALL COUNTER	
1510	2AC7	36 06 .	MVI M,6	
1511	2AC9	C1 . .	POP B ;RESTORE B,C	
1512	2ACA	2E 63 .	MVI L,UNIT0 ;HAS DIRECTION CHANGED?	
1513	2ACC	AE . .	XRA M	
1514	2ACD	E6 02 .	ANI FWD	
1515	2ACF	C8 . .	RZ ;NO - RETURN	
1516	2AD0	AE . .	XRA M ;YES - CHANGE LSTFWD	
1517	2AD1	77 . .	MOV M,A	
1518	2AD2	2E 5F .	MVI L,ABSTAK ;AND COMPLEMENT TACH COUNT	
1519	2AD4	7E . .	MOV A,M	
1520	2AD5	2F . .	CMA	
1521	2AD6	77 . .	MOV M,A	
1522	2AD7	23 . .	INX H	
1523	2AD8	7E . .	MOV A,M	
1524	2AD9	2F . .	CMA	
1525	2ADA	77 . .	MOV M,A	
1526	2ADB	C9 . .	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1528	2ADC	.	.	;	45
1529	2ADC	.	.	; * * * * *	
1530	2ADC	.	.	;	
1531	2ADC	.	.	; GTCTBT - GET BIT FOR SELECTED CTU	
1532	2ADC	.	.	;	
1533	2ADC	.	.	; ENTRY: DON'T CARE	
1534	2ADC	.	.	;	
1535	2ADC	.	.	; EXIT : A = BIT	
1536	2ADC	.	.	; Z => RGTCTU	
1537	2ADC	.	.	; NZ => LFTCTU	
1538	2ADC	.	.	;	
1539	2ADC	.	.	;	
1540	2ADC	.	.	GTCTBT EQU \$	
1541	2ADC	3A	55	LDA CMND ;WHICH UNIT IS SELECTED?	
1542	2ADF	E6	10	ANI USL ;1 => LEFT, 0 => RIGHT	
1543	2AE1	3E	01	MVI A,LFTCTU	
1544	2AE3	C0	.	RNZ ;LEFT - RETURN	
1545	2AE4	3E	02	MVI A,RGTCTU	
1546	2AE6	C9	.	RET	
1547	2AE7	.	.	;	
1548	2AE7	.	.	; * * * * *	
1549	2AE7	.	.	;	
1550	2AE7	.	.	; CHKFWD - CHECK DIRECTION OF TAPE MOTION	
1551	2AE7	.	.	;	
1552	2AE7	.	.	; EXIT : Z => REV; A = 0	
1553	2AE7	.	.	; NZ => FWD; A = FWD	
1554	2AE7	.	.	;	
1555	2AE7	.	.	CHKFWD EQU \$	
1556	2AE7	3A	55	LDA CMND	
1557	2AEA	E6	02	ANI FWD	
1558	2AEC	C9	.	RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 46
=====
1560     2AED      . . .      ;
1561     2AED      . . .      ; * * * * *
1562     2AED      . . .      ;
1563     2AED      . . .      ; CISCAN - CHECK FOR CARTRIDGE INSERTED
1564     2AED      . . .      ;
1565     2AED      . . .      ;     EXIT :  A,H,L DESTROYED
1566     2AED      . . .      ;             NC => TAPE INSERTED
1567     2AED      . . .      ;             C  => NO TAPE
1568     2AED      . . .      ;
1569     2AED      . . .      CISCAN EQU $
1570     2AED      CD 46 2A    CALL GETSTA      ;GET STATUS
1571     2AF0      67 . .     MOV H,A'        ;SAVE STATUS
1572     2AF1      CD DC 2A    CALL GTCTBT     ;GET BIT FOR SELECTED UNIT
1573     2AF4      A4 . .     ANA H           ;IS IT INSERTED?
1574     2AF5      C0 . .     RNZ             ;YES - RETURN
1575     2AF6      21 6E 3B    LXI H,NTPMSG   ;NO - REPORT ERROR
1576     2AF9      C3 E5 2D    JMP CTUERR
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1578	2AFC	.	.	;*****	47
1579	2AFC	.	.	; CHKEOF - CHECK FOR END OF FILE *	
1580	2AFC	.	.	;*****	
1581	2AFC	.	.	;	
1582	2AFC	.	.	; EXIT Z - NOT END OF FILE	
1583	2AFC	.	.	; A = 0	
1584	2AFC	.	.	; NZ - END OF FILE	
1585	2AFC	.	.	; A DESTROYED	
1586	2AFC	.	.	CHKEOF EQU \$	
1587	2AFC	3A	62 FF	LDA CNTRL0 ;GET LOGICAL STATUS	
1588	2AFF	E6	01 .	ANI EOF ;MASK FOR END OF FILE FLAG	
1589	2B01	C9	.	RET ;RETURN	
1590	2B02	.	.	;*****	
1591	2B02	.	.	; CHKEW - CHECK FOR TAPE PAST EARLY WARNING *	
1592	2B02	.	.	;*****	
1593	2B02	.	.	;	
1594	2B02	.	.	; EXIT NZ => TAPE PAST EARLY WARNING	
1595	2B02	.	.	; A DESTROYED	
1596	2B02	.	.	; H,L = ADDRESS OF END OF TAPE MESSAGE	
1597	2B02	.	.	; Z => NOT PAST EARLY WARNING	
1598	2B02	.	.	; A = 0	
1599	2B02	.	.	;	
1600	2B02	.	.	CHKEW EQU \$	
1601	2B02	3A	63 FF	LDA UNIT0	
1602	2B05	E6	80 .	ANI EW	
1603	2B07	C8	.	RZ	
1604	2B08	21	07 3B	LXI H,EOTMSG	
1605	2B0B	C3	E5 2D	JMP CTUERR	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 48
=====
1607      2B0E      . . .      ;*****
1608      2B0E      . . .      ; CHKEVO - CHECK FOR END-OF-VALID-DATA *
1609      2B0E      . . .      ;*****
1610      2B0E      . . .      ;
1611      2B0E      . . .      ; EXIT      NZ - END-OF-VALID-DATA REACHED
1612      2B0E      . . .      ;           A DESTROYED
1613      2B0E      . . .      ;           H,L = ADDRESS TO END-OF-DATA MESSAGE
1614      2B0E      . . .      ;           Z - NOT AT END-OF-VALID-DATA
1615      2B0E      . . .      ;           A = 0
1616      2B0E      . . .      ;
1617      2B0E      . . .      CHKEVO EQU $
1618      2B0E      3A 62 FF      LDA  CNTRL0      ;GET LOGICAL STATUS
1619      2B11      E6 02 .      ANI  EVD         ;END-OF-DATA REACHED?
1620      2B13      C8 . .      RZ              ;NO - RETURN
1621      2B14      21 1A 3B      LXI  H,EVDMSG   ;YES - SET END-OF-DATA MSG
1622      2B17      C9 . .      RET            ;RETURN
1623      2B18      . . .      ;*****
1624      2B18      . . .      ; CHKEVD - CHECK FOR END-OF-VALID-DATA OR *
1625      2B18      . . .      ;   LAST COMMAND WAS A RECORD OPERATION *
1626      2B18      . . .      ;*****
1627      2B18      . . .      ;
1628      2B18      . . .      ; EXIT      NZ => CONDITION TRUE
1629      2B18      . . .      ;           H,L = ADDRESS OF END OF DATA MESSAGE
1630      2B18      . . .      ;           Z => CONDITION NOT TRUE
1631      2B18      . . .      ;           A = 0
1632      2B18      . . .      ;
1633      2B18      . . .      CHKEVD EQU $
1634      2B18      3A 62 FF      LDA  CNTRL0
1635      2B18      E6 42 .      ANI  EVD+DATATR
1636      2B1D      C8 . .      RZ
1637      2B1E      21 1A 3B      LXI  H,EVDMSG
1638      2B21      C3 E5 2D      JMP  CTUERR
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
1640	2B24	.	.	;
1641	2B24	.	.	; * * * * *
1642	2B24	.	.	;
1643	2B24	.	.	STIOFS - SET FLAG(S) IN IOFLGS
1644	2B24	.	.	;
1645	2B24	.	.	ENTRY: A = BITS TO BE SET
1646	2B24	.	.	;
1647	2B24	.	.	EXIT : FLAGS SET, A = NEW IOFLGS
1648	2B24	.	.	H,L -> IOFLGS
1649	2B24	.	.	;
1650	2B24	.	.	STIOFS EQU \$
1651	2B24	21	65 FF	LXI H,IOFLGS
1652	2B27	B6	.	ORA M
1653	2B28	77	.	MOV M,A
1654	2B29	C9	.	RET
1655	2B2A	.	.	;
1656	2B2A	.	.	; * * * * *
1657	2B2A	.	.	;
1658	2B2A	.	.	CLIOFS - CLEAR FLAG(S) IN IOFLGS
1659	2B2A	.	.	;
1660	2B2A	.	.	ENTRY: A = -1-FLAGS TO BE CLEARED
1661	2B2A	.	.	;
1662	2B2A	.	.	EXIT : FLAGS CLEARED, A = NEW IOFLGS
1663	2B2A	.	.	Z => NO FLAGS CHANGED
1664	2B2A	.	.	NZ => AT LEAST ONE CLEARED
1665	2B2A	.	.	H,L -> IOFLGS
1666	2B2A	.	.	;
1667	2B2A	.	.	CLIOFS EQU \$
1668	2B2A	21	65 FF	LXI H,IOFLGS
1669	2B2D	A6	.	ANA M
1670	2B2E	BE	.	CMP M
1671	2B2F	77	.	MOV M,A
1672	2B30	C9	.	RET

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 50
1674	2B31	. . .	;*****	
1675	2B31	. . .	; HARDWARE DRIVERS *****	
1676	2B31	. . .	;*****	
1677	2B31	. . .	;	
1678	2B31	. . .	; * * * * *	
1679	2B31	. . .	;	
1680	2B31	. . .	; INTRWD - INTERRUPT DRIVEN REWIND	
1681	2B31	. . .	;	
1682	2B31	. . .	; ENTRY: H,L -> INTERRUPT ROUTINE TO BE	
1683	2B31	. . .	; USED AFTER BOT HOLES HAVE BEEN	
1684	2B31	. . .	; SEEN AND TAPE MOTION REVERSED	
1685	2B31	. . .	;	
1686	2B31	. . .	; EXIT : ALL REGISTERS DESTROYED	
1687	2B31	. . .	;	
1688	2B31	. . .	INTRWD EQU \$	
1689	2B31	E5 . .	PUSH H ;SAVE NEXT INTERRUPT ADDRESS	
1690	2B32	CD ED 2A	CALL CISCAN ;IS CARTRIDGE INSERTED?	
1691	2B35	D8 . .	RC ;NO - REPORT ERROR	
1692	2B36	3E 05 .	MVI A,5 ;SET SOFT COUNT TO 5	
1693	2B38	32 5D FF	STA SFTCNT	
1694	2B3B	CD 52 2D	CALL REVEVD ;RECORD EVD IF NEEDED	
1695	2B3E	E1 . .	POP H ;RECALL INTERRUPT ADDRESS	
1696	2B3F	D8 . .	RC ;RETURN ON ERROR	
1697	2B40	22 33 FF	SHLD CTIADR ;SAVE IN CTIADR	
1698	2B43	97 . .	SUB A ;CLEAR A	
1699	2B44	32 62 FF	STA CNTRL0 ;CLEAR CONTROL FLAGS	
1700	2B47	3C . .	INR A	
1701	2B48	32 5E FF	STA FILNUM ;INITIALIZE FILE COUNT	
1702	2B4B	3E 08 .	MVI A,CMDEXC ;SET SUCCESSFUL EXECUTION BI	
1703	2B4D	CD 9B 2B	CALL STUNTO	
1704	2B50	21 58 2B	LXI H,TIRWD ;GET INTERRUPT VECTOR	
1705	2B53	3E 05 .	MVI A,RUN+FST ;FAST REVERSE COMMAND	
1706	2B55	C3 8B 2A	JMP OCM000 ;ISSUE COMMAND AND EXIT	
1707	2B58	. . .	;	
1708	2B58	. . .	; INTERRUPT SERVICE ROUTINE	
1709	2B58	. . .	;	
1710	2B58	. . .	TIRWD EQU \$	
1711	2B58	E6 20 .	ANI BOT ;BEFORE BEGINNING OF TAPE?	
1712	2B5A	C0 . .	RNZ ;NO - CONTINUE WAITING	
1713	2B5B	CD E4 2B	CALL STOPTP ;YES - STOP THE TAPE	
1714	2B5E	2A 33 FF	LHLD CTIADR ;GET ADDRESS FOR NEXT INTR	
1715	2B61	3E 07 .	MVI A,RUN+FWD+FST	
1716	2B63	C3 8B 2A	JMP OCM000 ;ISSUE FAST FORWARD COMMAND	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 51
=====
1718      2866      . . .      ;
1719      2866      . . .      ; * * * * *
1720      2866      . . .      ;
1721      2866      . . .      ; RWDBOT - REWIND TO BEGINNING OF TAPE
1722      2866      . . .      ;
1723      2866      . . .      ; INTERRUPT DRIVEN
1724      2866      . . .      ; FIRST PART (CALL TO INTRWD) DESTROYS ALL RG
1725      2866      . . .      ; INTERRUPT ROUTINE DESTROYS NONE
1726      2866      . . .      ;
1727      2866      . . .      ;
1728      2866      . . .      RWDBOT EQU $
1729      2866      21 6C 2B      LXI H,TIRWBT ;GET INTERRUPT ROUTINE ADDR
1730      2869      C3 31 2B      JMP INTRWD   ;SET UP REWIND AND RETURN
1731      286C      . . .      ;
1732      286C      . . .      ; INTERRUPT ROUTINE - GO BACK OVER BOT HOLES
1733      286C      . . .      ; AND QUIT
1734      286C      . . .      ;
1735      286C      . . .      TIRWBT EQU $
1736      286C      E6 20 .      ANI BOT     ;AFTER BOT HOLES?
1737      286E      C8 . .      RZ          ;NO - CONTINUE WAITING
1738      286F      CD E4 2B      CALL STOPTP ;YES - STOP THE TAPE
1739      2872      C3 58 2A      JMP LITOFF  ;TURN OFF INDICATOR LIGHT
=====
  
```


13255

2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 53
=====
1784      2BA3      . . .      ;
1785      2BA3      . . .      ; THIRD INTERRUPT ROUTINE - GAP OUT LEADER
1786      2BA3      . . .      ;
1787      2BA3      . . .      TIRLP2 EQU $
1788      2BA3      7E . .      MOV A,M      ;TEST TAPE STATUS
1789      2BA4      87 . .      ORA A
1790      2BA5      F0 . .      RP          ;IF NO TACH, RETURN
1791      2BA6      2E 61 .      MVI L,RELTAK ;INCREMENT COUNTER
1792      2BA8      34 . .      INR M       ;=0?
1793      2BA9      C0 . .      RNZ        ;NO - RETURN
1794      2BAA      C3 E4 2B      JMP STOPTH  ;YES - STOP TAPE AND EXIT
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 54
=====
1796      2BAD      . . .      ;
1797      2BAD      . . .      ; * * * * *
1798      2BAD      . . .      ;
1799      2BAD      . . .      ;      CHKLPM - MOVE TO LOAD POINT IF BEFORE LP
1800      2BAD      . . .      ;
1801      2BAD      . . .      ;      ENTRY:  DON'T CARE
1802      2BAD      . . .      ;
1803      2BAD      . . .      ;      EXIT :  A,H,L DESTROYED
1804      2BAD      . . .      ;              C => ERROR
1805      2BAD      . . .      ;              NC => NO ERROR
1806      2BAD      . . .      ;              NZ => ALREADY THERE
1807      2BAD      . . .      ;              Z  => MOVED THERE
1808      2BAD      . . .      CHKLPM EQU $
1809      2BAD      3A 63 FF    LDA UNIT0      ;BEFORE LP?
1810      2BB0      E6 40 .     ANI LP
1811      2BB2      C0 . .     RNZ              ;NO - RETURN
1812      2BB3      CD ED 2A    CALL CISCAN     ;SELECTED TAPE INSERTED?
1813      2BB6      D8 . .     RC              ;NO - RETURN ERROR
1814      2BB7      CD 7E 2B    CALL RLP100    ;START THE TAPE
1815      2BBA      21 DF 3A    LXI H,LLPMSG   ;"LOCATING LOAD POINT"
1816      2BBD      CD A3 3C    CALL CARDIO
1817      2BC0      . . .      LPM010 EQU $    ;WAIT FOR TAPE TO STOP
1818      2BC0      CD 2E 48    CALL RETSCN    ;LOOK FOR RETURN KEY
1819      2BC3      D4 C1 29    CNC CTMON1     ;NONE - IS TAPE FINISHED?
1820      2BC6      C2 C0 2B    JNZ LPM010     ;TAPE STILL RUNNING - WAIT
1821      2BC9      C3 43 00    JMP RSTDSP     ;TAPE FINISHED - RESTORE DIS
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1823	2BCC	.	.	;	55
1824	2BCC	.	.	; * * * * *	
1825	2BCC	.	.	;	
1826	2BCC	.	.	; ENDBAK - CONDITION TAPE ON SELECTED DRIVE	
1827	2BCC	.	.	; (WIND OUT TO END-OF-TAPE AND BACK)	
1828	2BCC	.	.	;	
1829	2BCC	.	.	; THIS IS AN INTERRUPT DRIVEN ROUTINE.	
1830	2BCC	.	.	; FIRST PART DESTROYS ALL REGISTERS.	
1831	2BCC	.	.	; INTERRUPT ROUTINE DESTROYS NONE.	
1832	2BCC	.	.	;	
1833	2BCC	.	.	;	
1834	2BCC	.	.	; ENDBAK EQU \$	
1835	2BCC	21	02 2B	LXI H,TIEBK ;GET NEXT INTERRUPT ROUTINE	
1836	2BCF	C3	31 2B	JMP INTRWD ;GO TO REWIND ROUTINE	
1837	2BD2	.	.	;	
1838	2BD2	.	.	; INTERRUPT ROUTINE	
1839	2BD2	.	.	; WATCH 'TRIAL' SINGLE HOLES GO BY, THEN	
1840	2BD2	.	.	; REWIND	
1841	2BD2	.	.	;	
1842	2BD2	.	.	; TIEBK EQU \$	
1843	2BD2	E6	80 .	ANI EW ;BEYOND EARLY WARNING YET?	
1844	2BD4	C8	.	RZ ;NO - CONTINUE WAITING	
1845	2BD5	CD	E4 2B	CALL STOPTP ;YES - STOP THE TAPE	
1846	2BD8	C5	.	PUSH B	
1847	2BD9	D5	.	PUSH D	
1848	2BDA	CD	66 2B	CALL RWDBOT ;SET UP THE REWIND	
1849	2BDD	D1	.	POP D ;PREPARE TO EXIT	
1850	2BDE	C1	.	POP B	
1851	2BDF	C9	.	RET ;DONE - RWDBOT HANDLES REST	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1853	2BE0	.	.	;	56
1854	2BE0	.	.	; * * * * *	
1855	2BE0	.	.	;	
1856	2BE0	.	.	; STOPTP - STOP THE TAPE	
1857	2BE0	.	.	;	
1858	2BE0	.	.	; ENTRY: DON'T CARE	
1859	2BE0	.	.	;	
1860	2BE0	.	.	; EXIT : TAKPO INC'D BY TACH EDGES IN GAP	
1861	2BE0	.	.	; CTIVEC -> "DO NOTHING" ROUTINE	
1862	2BE0	.	.	; INTERRUPTS ENABLED	
1863	2BE0	.	.	; Z (REQ'D BY PUTCTU)	
1864	2BE0	.	.	; NC (REQ'D BY RWDBOT)	
1865	2BE0	.	.	; A,H,L DESTROYED	
1866	2BE0	.	.	;	
1867	2BE0	.	.	; ENTRY STPTP0: CLEAR TACH COUNT FIRST	
1868	2BE0	.	.	; ENTRY STPTP1: USE INTERRUPT VECTOR IN H,L	
1869	2BE0	.	.	;	
1870	2BE0	.	.	;	
1871	2BE0	.	.	STPTP0 EQU \$	
1872	2BE0	97	.	SUB A	
1873	2BE1	32	61 FF	STA RELTAK	
1874	2BE4	.	.	STOPTP EQU \$	
1875	2BE4	21	FE 2B	LXI H,TISTOP ;SET UP STOP INTR ROUTINE	
1876	2BE7	.	.	STPTP1 EQU \$	
1877	2BE7	3A	55 FF	LDA CMND ;RE-ISSUE COMMAND W/O RUN BI	
1878	2BEA	E6	FE .	ANI -1-RUN	
1879	2BEC	CD	8B 2A	CALL OCM000	
1880	2BEF	FB	.	EI ;ENABLE INTERRUPTS	
1881	2BF0	.	.	STP100 EQU \$;WAIT FOR TAPE TO STOP	
1882	2BF0	3A	50 FF	LDA TPSTAL ;REQUIRES 50 MSEC W/O A CTU	
1883	2BF3	B7	.	ORA A ;INTERRUPT	
1884	2BF4	C2	F0 2B	JNZ STP100	
1885	2BF7	21	7D 2B	LXI H,TID00 ;TAPE STOPPED	
1886	2BFA	22	E1 FF	SHLD CTIVEC ;SET UP "DO NOTHING" INT ROU	
1887	2BFD	C9	.	RET	
1888	2BFE	.	.	;*****	
1889	2BFE	.	.	; "STOP TAPE" INTERRUPT ROUTINE *	
1890	2BFE	.	.	;*****	
1891	2BFE	.	.	TISTOP EQU \$	
1892	2BFE	3A	51 FF	LDA HOLCNT ;CHECKING OUT HOLE?	
1893	2C01	B7	.	ORA A	
1894	2C02	2E	55 .	MVI L,CMND ;(GET COMMAND)	
1895	2C04	7E	.	MOV A,M	
1896	2C05	1F	.	RAK ;(CLEAR "RUN" BIT)	
1897	2C06	07	.	RLC	
1898	2C07	CA	0C 2C	JZ STP200 ;NO - USE STOP COMMAND	
1899	2C0A	F6	01 .	ORI RUN ;YES - USE RUN COMMAND	
1900	2C0C	.	.	STP200 EQU \$	
1901	2C0C	BE	.	CMP M ;DESIRED COMMAND ISSUED?	
1902	2C0D	C4	8E 2A	CNZ OCM001 ;NO - ISSUE IT	

13255
2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 57
=====
1903     2C10     2E 66 .      MVI L,CTSTAT ;"DO NOTHG" WANTS H,L->CTSTA
1904     2C12     C3 7D 28     JMP TID00    ;INC TACH-GAP COUNT & QUIT
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1906	2C15	.	.	;	58
1907	2C15	.	.	; * * * * *	
1908	2C15	.	.	;	
1909	2C15	.	.	; GAPTST - TEST GAP LENGTH FOR FILE MARK	
1910	2C15	.	.	;	
1911	2C15	.	.	; CALLED BY RECORD SPACE AND FILE SPACE --	
1912	2C15	.	.	; INCREMENTS FILNUM BASED ON LENGTH OF LAST	
1913	2C15	.	.	; GAP PROCESSED (GAP LENGTH IN RELTAK).	
1914	2C15	.	.	;	
1915	2C15	.	.	; EXIT : A,H,L DESTROYED	
1916	2C15	.	.	; NC => FILNUM CHANGED	
1917	2C15	.	.	; ' A = 0 => DECREMENTED	
1918	2C15	.	.	; A # 0 => INCREMENTED	
1919	2C15	.	.	; C => FILNUM UNCHANGED	
1920	2C15	.	.	;	
1921	2C15	.	.	;	
1922	2C15	.	.	GAPTST EQU \$	
1923	2C15	21	62	FF LXI H,CNTRL0 ;GET PTR TO TAPE FLAGS	
1924	2C18	7E	.	MOV A,M	
1925	2C19	E6	02	ANI EVD ;BACKED OVER EVD?	
1926	2C18	C2	41	2C JNZ GPT030 ;YES - MARK STATUS	
1927	2C1E	3A	61	FF LDA RELTAK ;GET LENGTH OF LAST GAP	
1928	2C21	FE	41	CPI 65 ;INTER-FILE MARK GAP?	
1929	2C23	D2	35	2C JNC GPT010 ;YES - ADJUST FILE COUNT	
1930	2C26	FE	23	CPI 35 ;NO - FILE MARK GAP?	
1931	2C28	D8	.	RC ;NO - RETURN	
1932	2C29	7E	.	MOV A,M ;YES - CHANGE EOF STATUS	
1933	2C2A	EE	01	XRI EOF	
1934	2C2C	77	.	MOV M,A	
1935	2C2D	07	.	RLC ;EOF AND CMND[FWD] BOTH ON 0	
1936	2C2E	2E	55	MVI L,CMND-BASE ;BOTH OFF => NEW FILE	
1937	2C30	AE	.	XRA M ;ENTERED	
1938	2C31	E6	02	ANI FWD	
1939	2C33	37	.	STC	
1940	2C34	C0	.	RNZ	
1941	2C35	.	.	GPT010 EQU \$;ADJUST FILE NUMBER	
1942	2C35	2E	5E	MVI L,FILNUM	
1943	2C37	CD	E7	2A CALL CHKFWD ;GOING FORWARD?	
1944	2C3A	CA	3F	2C JZ GPT020	
1945	2C3D	34	.	INR M ;YES - INCREMENT FILE COUNT	
1946	2C3E	C9	.	RET	
1947	2C3F	.	.	GPT020 EQU \$	
1948	2C3F	35	.	DCR M ;NO - DECREMENT FILE COUNT	
1949	2C40	C9	.	RET	
1950	2C41	.	.	GPT030 EQU \$	
1951	2C41	36	01	MVI M,EOF ;BACKED OVER EVD - SET EOF	
1952	2C43	37	.	STC ;AND CLEAR EVD	
1953	2C44	C9	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1955	2C45	.	.	;	59
1956	2C45	.	.	; * * * * *	
1957	2C45	.	.	;	
1958	2C45	.	.	; FWDSPC - SPACE FORWARD N RECORDS, RETURN	
1959	2C45	.	.	; WHEN SPACING IS FINISHED	
1960	2C45	.	.	; FWDSPX - SAME, BUT RET WHEN SPACING STARTED	
1961	2C45	.	.	; FWDSP1 - CLEARS RELTAK AND SPACES ONE	
1962	2C45	.	.	; CHECKS EVD, BUT NOT DATA RECORDED(DATATR)	
1963	2C45	.	.	;	
1964	2C45	.	.	; BAKSPW - SPACE BACK N RECORDS, POSITIONING	
1965	2C45	.	.	; TAPE FOR WRITE (I.E., RELTAK = LENGTH OF	
1966	2C45	.	.	; GAP FROM LAST RECORD) RETURN WHEN FINISHD	
1967	2C45	.	.	; BAKSPX - SAME, BUT RET WHEN SPACING STARTED	
1968	2C45	.	.	; IF LAST RECORD IS EOF, GO FORWARD OVER IT	
1969	2C45	.	.	; BAKSPR - SPACE BACK N RECORDS, POSITIONING	
1970	2C45	.	.	; TAPE FOR READ (I.E., DON'T WORRY ABOUT	
1971	2C45	.	.	; GAP LENGTHS)	
1972	2C45	.	.	;	
1973	2C45	.	.	; WHEN IN DOUBT, USE BAKSPW!!!!	
1974	2C45	.	.	;	
1975	2C45	.	.	; ENTRY: A = NUMBER OF RECORDS TO SPACE	
1976	2C45	.	.	; UNIT SELECTED AND NOT RUNNING	
1977	2C45	.	.	; H,L = NUMBER OF RECORDS	
1978	2C45	.	.	;	
1979	2C45	.	.	; EXIT : C => ERROR	
1980	2C45	.	.	; NC => NO ERROR	
1981	2C45	.	.	; A,H,L DESTROYED	
1982	2C45	.	.	;	
1983	2C45	.	.	FWDSP1 EQU \$	
1984	2C45	97	.	SUB A ;CLEAR GAP-TACH COUNTER	
1985	2C46	32	61	STA RELTAK	
1986	2C49	21	01	LXI H,1 ;SET FOR ONE RECORD	
1987	2C4C	CD	57	CALL FWDSPX ;START SPACING	
1988	2C4F	D8	.	RC ;RETURN IF ERROR	
1989	2C50	.	.	FSP020 EQU \$;(ENTRY FOR BACKSPACE)	
1990	2C50	CD	C1	CALL CTMON1 ;MONITOR TAPES	
1991	2C53	C2	50	JNZ FSP020 ;LOOP IF TAPE RUNNING	
1992	2C56	C9	.	RET ;OTHERWISE, RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 60
=====
1994      2C57      . . .      FWDSPX EQU $
1995      2C57      7C . .      MOV A,H
1996      2C58      B5 . .      ORA L ;COUNT = 0?
1997      2C59      C8 . .      RZ ;YES - RETURN
1998      2C5A      22 2D FF    SHLD CTICNT+1 ;NO - STORE COUNT IN CTICNT+
1999      2C5D      97 . .      SUB A ;AND CLEAR "MEASURE GAP"
2000      2C5E      32 2C FF    STA CTICNT ;FLAG
2001      2C61      CD AD 2B    CALL CHKLPM ;MOVE TAPE TO LP IF BEFORE L
2002      2C64      D8 . .      RC ;RETURN ON ERROR
2003      2C65      CD 0E 2B    CALL CHKEVO ;AT END OF DATA?
2004      2C68      C2 E5 2D    JNZ CTUERR ;YES - QUIT
2005      2C6B      CD 02 2B    CALL CHKEW ;NO - PAST EARLY WARNING?
2006      2C6E      D8 . .      RC ;YES - RETURN ERROR
2007      2C6F      3E 03 .     MVI A,RUN+FWD ;SLOW FORWARD COMMAND
2008      2C71      C3 A2 2C    JMP RECSPC ;GO TO SPACING ROUTINE
2009      2C74      . . .      BAKSPW EQU $
2010      2C74      3E FF .     MVI A,-1 ;SET "MEASURE GAP" FLAG
2011      2C76      C3 7B 2C    JMP BSP010
2012      2C79      . . .      BAKSPR EQU $
2013      2C79      3E 00 .     MVI A,0 ;CLEAR "MEASURE GAP" FLAG
2014      2C7B      . . .      BSP010 EQU $
2015      2C7B      CD 8D 2C    CALL BSP020 ;START SPACING
2016      2C7E      C3 50 2C    JMP FSP020
2017      2C81      . . .      BAKSPX EQU $
2018      2C81      7D . .      MOV A,L
2019      2C82      3D . .      DCR A ;SKIP ONE RECORD ONLY?
2020      2C83      B4 . .      ORA H ;(ADD IN MSB)
2021      2C84      C2 8B 2C    JNZ BSP015 ;NO - CONTINUE
2022      2C87      CD FC 2A    CALL CHKEOF ;AT END OF FILE?
2023      2C8A      C0 . .      RNZ ;YES - RETURN AT ONCE
2024      2C8B      . . .      BSP015 EQU $
2025      2C8B      3E 01 .     MVI A,1 ;SET "BACK OVER EOF" FLAG
2026      2C8D      . . .      BSP020 EQU $
2027      2C8D      32 2C FF    STA CTICNT ;SET SKIP FLAG
2028      2C90      7C . .      MOV A,H ;SKIP COUNT NON-ZERO?
2029      2C91      B5 . .      ORA L
2030      2C92      C8 . .      RZ ;NO - RETURN
2031      2C93      22 2D FF    SHLD CTICNT+1 ;YES - STORE COUNT
2032      2C96      3A 63 FF    LDA UNITO ;BEFORE LP?
2033      2C99      E6 40 .     ANI LP
2034      2C9B      C8 . .      RZ ;YES - RETURN
2035      2C9C      97 . .      SUB A ;NO - CLEAR GAP COUNTER
2036      2C9D      32 61 FF    STA RELTAK
2037      2CA0      3E 01 .     MVI A,RUN ;SLOW BACK COMMAND
2038      2CA2      . . .      ; FALL INTO SPACING ROUTINE
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2040	2CA2	.	.	;	61
2041	2CA2	.	.	; SPACE OVER N RECORDS	
2042	2CA2	.	.	;	
2043	2CA2	.	.	RECSPC EQU \$	
2044	2CA2	2A	2D	LHLD CTICNT+1 ;GET # OF RECORDS TO SKIP	
2045	2CA5	C5	.	PUSH B ;(SAVE B,C)	
2046	2CA6	01	F4	LXI B,-12 ;MORE THAN 12 RECORDS TO	
2047	2CA9	09	.	DAD B ;SKIP?	
2048	2CAA	C1	.	POP B ;(RESTORE B,C AND SET	
2049	2CAB	21	B6	LXI H,TIRSP0 ;INTR TO WAIT FOR DATA)	
2050	2CAE	D2	78	JNC OUTCMD ;NO - ISSUE COMMAND AND EXIT	
2051	2CB1	F6	04	ORI FST ;YES - USE HIGH SPEED SKIP	
2052	2CB3	C3	78	JMP OUTCMD ;ISSUE COMMAND AND EXIT	
2053	2CB6	.	.	;	
2054	2CB6	.	.	; WAIT FOR DATA - INTERRUPT ROUTINE	
2055	2CB6	.	.	;	
2056	2CB6	.	.	TIRSP0 EQU \$	
2057	2CB6	E6	40	ANI LP ;BEFORE LP?	
2058	2CB8	CA	E0	JZ STPTP0 ;YES - QUIT	
2059	2CBB	7E	.	MOV A,M ;GET STATUS	
2060	2CBC	E6	20	ANI GAP ;STILL IN GAP?	
2061	2CBE	C2	85	JNZ SRC600 ;YES - CHECK FOR EVD	
2062	2CC1	.	.	;	
2063	2CC1	.	.	; DATA FOUND - EVALUATE GAP	
2064	2CC1	.	.	;	
2065	2CC1	CD	15	CALL GAPTST ;CHECK GAP TYPE, AND UPDATE	
2066	2CC4	.	.	EOF STATUS AND FILE NUMBER	
2067	2CC4	.	.	RSP040 EQU \$;(ENTRY FOR FWD OVER EOF)	
2068	2CC4	97	.	SUB A ;CLEAR GAP COUNTER	
2069	2CC5	32	61	STA RELTAK	
2070	2CC8	2A	2D	LHLD CTICNT+1 ;BACKSPACING OVER FINAL GAP?	
2071	2CCB	7C	.	MOV A,H ;(COUNT = 0 => YES)	
2072	2CCC	B5	.	ORA L	
2073	2CCD	21	E4	LXI H,TIRSP1 ;NO - SET INTERRUPT TO WAIT	
2074	2CD0	22	E1	SHLD CTIVEC ;FOR GAP	
2075	2CD3	C0	.	RNZ ;AND EXIT	
2076	2CD4	CD	E4	CALL STOPTP ;YES - STOP THE TAPE	
2077	2CD7	3E	01	MVI A,1 ;SET COUNT TO 1	
2078	2CD9	32	2D	STA CTICNT+1	
2079	2CDC	21	B6	LXI H,TIRSP0 ;WAIT FOR DATA INT ROUT	
2080	2CDF	3E	03	MVI A,RUN+FWD ;START SPCING FORWARD	
2081	2CE1	C3	8B	JMP OCM000	
2082	2CE4	.	.	;	
2083	2CE4	.	.	; WAIT FOR GAP - INTERRUPT ROUTINE	
2084	2CE4	.	.	;	
2085	2CE4	.	.	TIRSP1 EQU \$	
2086	2CE4	E6	40	ANI LP ;BEFORE LP?	
2087	2CE6	CA	E0	JZ STPTP0 ;YES - QUIT	
2088	2CE9	3A	20	LDA IOCTDI ;CLEAR POSSIBLE BYTE READY	
2089	2CEC	7E	.	MOV A,M ;GET STATUS	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 62
=====
2090     2CED     E6  20  .      ANI  GAP          ;IN GAP YET?
2091     2CEF     2E  61  .      MVI  L,RELTAK     ;(GET POINTER TO TACH)
2092     2CF1     C2  F7  2C     JNZ  RSP110
2093     2CF4     36  00  .      MVI  M,0          ;NO - CLEAR GAP LENGTH COUNT
2094     2CF6     C9  .   .      RET              ;AND RETURN
2095     2CF7     .   .   .      RSP110 EQU $      ;YES -
2096     2CF7     34  .   .      INR  M           ;INCREMENT GAP LENGTH COUNT
2097     2CF8     7E  .   .      MOV  A,M         ;(GAP >= 3 - ASSUME GAP)
2098     2CF9     FE  03  .      CPI  3          ;(GAP < 3 - ASSUME DROPOUT
2099     2CFB     D8  .   .      RC           ;GAP NOT YET 3 - RETURN
2100     2CFC     21  B6  2C     LXI  H,TIRSP0    ;GAP >= 3 - SET INTERRUPT TO
2101     2CFF     22  E1  FF     SHLD CTIVEC      ;WAIT FOR DATA
2102     2D02     2A  2D  FF     LHLD CTICNT+1   ;GET SKIP COUNTER
2103     2D05     2B  .   .      DCX  H          ;DECREMENT COUNT
2104     2D06     22  2D  FF     SHLD CTICNT+1   ;UPDATE COUNTER
2105     2D09     7C  .   .      MOV  A,H
2106     2D0A     B7  .   .      ORA  A          ;SKIP COMPLETED?
2107     2D0B     C0  .   .      RNZ          ;NO - CONTINUE SKIPPING
2108     2D0C     B5  .   .      ORA  L          ;BOTH MSB AND LSB = 0?
2109     2D0D     CA  1B  2D     JZ   RSP120     ;YES - TERMINATE SKIP
2110     2D10     FE  0C  .      CPI  12         ;LESS THAN 12 MORE TO GO?
2111     2D12     D0  .   .      RNC          ;NO - CONTINUE HIGH SPEED
2112     2D13     3A  55  FF     LDA  CMND       ;YES - CLEAR THE FAST BIT
2113     2D16     E6  FB  .      ANI  -1-FST     ;TO START SLOW SKIP
2114     2D18     C3  8E  2A     JMP  OCM001     ;OUTPUT THE COMMAND AND EXIT
2115     2D1B     .   .   .      ;*****
2116     2D1B     .   .   .      ; SKIP COMPLETED - CHECK TERMINATION TYPE *
2117     2D1B     .   .   .      ;*****
2118     2D1B     .   .   .      RSP120 EQU $
2119     2D18     21  2C  FF     LXI  H,CTICNT   ;GET SKIP FLAG
2120     2D1E     B6  .   .      ORA  M          ;WHAT TYPE ENDING?
2121     2D1F     36  00  .      MVI  M,0        ;(CLEAR SKIP FLAG)
2122     2D21     CA  E4  2B     JZ   STOPTH     ;NO ACTION - STOP TAPE & RET
2123     2D24     F8  .   .      RM           ;BACK TO WRITE - SPC OVER GA
2124     2D25     CD  FC  2A     CALL CHKEOF     ;AT END OF FILE?
2125     2D28     C2  C4  2C     JNZ  RSP040     ;YES - SPACE OVER FILE MARK
2126     2D2B     3E  80  .      MVI  A,E0FINH   ;NO - INHIBIT END OF FILE
2127     2D2D     C3  72  32     JMP  SETCT0     ;REPORT AND EXIT
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 63
2129	2030	.	.	.	;
2130	2030	.	.	.	; * * * * *
2131	2030	.	.	.	;
2132	2030	.	.	.	; EVDBSP - BACKSPACE OVER EVD IF NEEDED
2133	2030	.	.	.	;
2134	2030	.	.	.	; ENTRY: DON'T CARE
2135	2030	.	.	.	;
2136	2030	.	.	.	; EXIT : A,H,L DESTROYED
2137	2030	.	.	.	; C => TAPE REMOVED
2138	2030	.	.	.	; NC => NO ERROR
2139	2030	.	.	.	; TAPE STOPPED
2140	2030	.	.	.	;
2141	2030	.	.	.	;
2142	2030	.	.	.	; EVDBSP EQU \$
2143	2030	CD	0E	2B	; CALL CHKEVO ; AT END OF DATA MARK?
2144	2033	C8	.	.	; RZ ; NO - RETURN
2145	2034	21	01	00	; LXI H,1 ; YES - BACKSPACE ONE RECORD
2146	2037	CD	74	2C	; CALL BAKSPW ; BAKSPW WAITS UNTIL FINISHED
2147	203A	C3	AD	2B	; JMP CHKLPM ; GO FWD IF EVD WAS ONLY THIN
2148	203D	.	.	.	; ON TAPE

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 64
=====
2150      2D3D      . . .      ;*****
2151      2D3D      . . .      ; EOFC - RECORD AN EOF MARK ON TAPE *
2152      2D3D      . . .      ;*****
2153      2D3D      . . .      EOFC      EQU $
2154      2D3D      CD FF 3C    CALL GTIOB0    ;GET A BUFFER
2155      2D40      D8 . .      RC              ;RETURN ON ERROR
2156      2D41      CD DC 2A    CALL GTCTBT    ;YES - GET BIT FOR THIS UNIT
2157      2D44      77 . .      MOV M,A        ;MARK BUFFER BUSY
2158      2D45      47 . .      MOV B,A        ;BUF2CT WANTS BIT IN B-REG
2159      2D46      2B . .      DCX H          ;H,L -> TYPE
2160      2D47      36 00 .     MVI M,0        ;MARK EOF
2161      2D49      2B . .      DCX H          ;MARK LENGTH = 1
2162      2D4A      36 01 .     MVI M,1
2163      2D4C      23 . .      INX H
2164      2D4D      23 . .      INX H
2165      2D4E      EB . .      XCHG           ;BUF2CT WANTS D,E -> STATUS
2166      2D4F      C3 F7 2F    JMP BUF2CT     ;BUF2CT HANDLES REST
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 65
2168	2052	.	.	*****	
2169	2052	.	.	; REVEVD - RECORD AN EVD ON A REVERSE TAPE *	
2170	2052	.	.	; COMMAND AFTER A RECORD OPERATION *	
2171	2052	.	.	;	
2172	2052	.	.	; ENTRY: B = INCREMENT FOR "IOCCNT"	
2173	2052	.	.	; IF EVD NEEDED	
2174	2052	.	.	;	
2175	2052	.	.	; EXIT : ALL REGISTERS DESTROYED	
2176	2052	.	.	;	
2177	2052	.	.	REVEVD EQU \$	
2178	2052	3A	62 FF	LDA CNTRL0 ;GET TAPE STATUS	
2179	2055	E6	40 .	ANI DATATR ;WAS LAST CMND RECORD?	
2180	2057	C8	. .	RZ ;NO - RETURN	
2181	2058	21	D5 FF	LXI H,IOCCNT	
2182	2058	7E	. .	MOV A,M ;FETCH SPACE COUNT	
2183	205C	80	. .	ADD B ;ADD IN INCREMENT	
2184	205D	77	. .	MOV M,A ;STORE NEW COUNT	
2185	205E	.	.	EVDWAT EQU \$;ENTRY TO WRITE EVD AND WAIT	
2186	205E	CD	69 2D	CALL EVDC ;START RECORDING EVD	
2187	2061	D8	. .	RC ;RETURN ON ERROR	
2188	2062	.	.	REV010 EQU \$	
2189	2062	CD	C1 29	CALL CTMON1 ;TAPE STILL MOVING?	
2190	2065	C2	62 2D	JNZ REV010 ;YES - WAIT	
2191	2068	C9	. .	RET ;NO - RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 66
=====
2193      2D69      . . .      ;*****
2194      2D69      . . .      ; EVDC - RECORD AN EVD MARK ON TAPE *
2195      2D69      . . .      ;*****
2196      2D69      . . .      EVDC EQU $ ;WAIT FOR A BUFFER
2197      2D69      CD FF 3C    CALL GTIOB0 ;GET A BUFFER
2198      2D6C      D8 . .     RC ;RETURN ON ERROR
2199      2D6D      CD DC 2A    CALL GTCTBT ;GET BIT FOR SELECTED UNIT
2200      2D70      77 . .     MOV M,A ;MARK BUFFER BUSY
2201      2D71      47 . .     MOV B,A ;BUF2CT TAKES UNIT IN B
2202      2D72      2B . .     DCX H ;BUF FREE - H,L -> TYPE
2203      2D73      36 01 .     MVI M,1 ;MARK EVD
2204      2D75      23 . .     INX H ;H,L -> STATUS
2205      2D76      EB . .     XCHG ;BUF2CT TAKES BUF PTR IN D,E
2206      2D77      C3 F7 2F    JMP BUF2CT ;START THE WRITE
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
2208	2D7A	.	.	;	67
2209	2D7A	.	.	; * * * * *	
2210	2D7A	.	.	;	
2211	2D7A	.	.	; FILCMP - COMPARE FILNUM AND IOCCNT	
2212	2D7A	.	.	;	
2213	2D7A	.	.	; ENTRY: DON'T CARE	
2214	2D7A	.	.	;	
2215	2D7A	.	.	; EXIT : Z => SAME	
2216	2D7A	.	.	; NZ => DIFFERENT	
2217	2D7A	.	.	; C => PRESENT < NEEDED	
2218	2D7A	.	.	; NC => PRESENT > NEEDED	
2219	2D7A	.	.	;	
2220	2D7A	.	.	;	
2221	2D7A	.	.	FILCMP EQU \$	
2222	2D7A	CD	E7 2A	CALL CHKFWD ;GOING FORWARD?	
2223	2D7D	.	.	FILCM1 EQU \$;(ENTRY TO AVOID FWD CHECK	
2224	2D7D	3A	5E FF	LDA FILNUM ;(GET PRESENT FILE COUNT)	
2225	2D80	C2	84 2D	JNZ FCP100 ;YES - USE ACTUAL FILE COUNT	
2226	2D83	3C	.	INR A ;NO - TARGET IS PREVIOUS FIL	
2227	2D84	.	.	FCP100 EQU \$	
2228	2D84	21	2D FF	LXI H,CTICNT+1 ;PTR TO NEEDED FILE COUNT	
2229	2D87	BE	.	CMP M ;COMPARE	
2230	2D88	C9	.	RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 68
=====
2232      2D89      . . .      ;*****
2233      2D89      . . .      ; LOGICAL DRIVERS *****
2234      2D89      . . .      ;*****
2235      2D89      . . .      ;*****
2236      2D89      . . .      ; SELECT UNIT INDICATED IN A-REG *
2237      2D89      . . .      ;*****
2238      2D89      . . .      SELACT EQU $
2239      2D89      0F . . .      RRC ;RIGHT UNIT SELECTED?
2240      2D8A      D2 BA 2D . . .      JNC SELRCT ;YES - GO GET IT
2241      2D8D      . . .      ;*****
2242      2D8D      . . .      ; SELCT - SELECT LEFT UNIT *
2243      2D8D      . . .      ;*****
2244      2D8D      . . .      SELCT EQU $
2245      2D8D      CD DC 2A . . .      CALL GTCTBT ;UNIT ZERO (LEFT) SELECTED?
2246      2D90      C0 . . .      RNZ ;YES - DON'T SWAP VARIABLES
2247      2D91      . . .      SELOPP EQU $
2248      2D91      C5 . . .      PUSH B ;SAVE REGISTERS
2249      2D92      D5 . . .      PUSH D
2250      2D93      21 5D FF . . .      LXI H,SFTCNT ;BOTTOM OF ACTIVE LIST
2251      2D96      11 56 FF . . .      LXI D,OTHER ;BOTTOM OF RESERVE LIST
2252      2D99      0E 07 . . .      MVI C,SFTCNT-OTHER ;VARIABLE COUNT
2253      2D98      . . .      XCH050 EQU $
2254      2D98      46 . . .      MOV B,M ;GET ACTIVE VAR.
2255      2D9C      1A . . .      LDAX D ;GET RESERVE VARIABLE
2256      2D9D      77 . . .      MOV M,A ;RESTORE RESERVE VARIABLE
2257      2D9E      78 . . .      MOV A,B ;SAVE ACTIVE VARIABLE
2258      2D9F      12 . . .      STAX D
2259      2DA0      23 . . .      INX H ;INCREMENT ACTIVE POINTER
2260      2DA1      13 . . .      INX D ;INCREMENT RESERVE POINTER
2261      2DA2      0D . . .      DCR C ;HAVE ALL VAR. BEEN SWAPPED?
2262      2DA3      C2 98 2D . . .      JNZ XCH050 ;NO - KEEP GOING
2263      2DA6      2E 55 . . .      MVI L,CMND
2264      2DA8      F3 . . .      DI ;STAY OUT OF INTERRUPT ROUT
2265      2DA9      7E . . .      MOV A,M
2266      2DAA      EE 10 . . .      XRI USL ;SWITCH UNITS
2267      2DAC      77 . . .      MOV M,A
2268      2DAD      32 00 8B . . .      STA IOCTCO ;ISSUE COMMAND
2269      2DB0      3A 00 8B . . .      LDA IOCTSI ;CLEAR ANY BAD INTERRUPTS
2270      2DB3      3A 20 8B . . .      LDA IOCTDI
2271      2DB6      FB . . .      EI ;(CANNOT CHANGE UNITS UNDER
2272      2DB7      . . .      ; INTRRUPT)
2273      2DB7      D1 . . .      POP D ;RESTORE REGISTERS
2274      2DB8      C1 . . .      POP B
2275      2DB9      C9 . . .      RET
2276      2DBA      . . .      ;*****
2277      2DBA      . . .      ; SELRCT - SELECT RIGHT UNIT *
2278      2DBA      . . .      ;*****
2279      2DBA      . . .      SELRCT EQU $
2280      2DBA      CD DC 2A . . .      CALL GTCTBT ;UNIT ONE (RIGHT) SELECTED?
2281      2DBD      C8 . . .      RZ ;YES - DON'T SWAP VARIABLES
=====

```


13255
2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                PAGE 69
=====
2282     20BE     C3  91  2D          JMP SELOPP      ;SWAP VARIABLES
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 70
=====
2284      2DC1      . . .      ;
2285      2DC1      . . .      ; * * * * *
2286      2DC1      . . .      ;
2287      2DC1      . . .      ;          BSYCHK - CHECK IF CTU BUSY
2288      2DC1      . . .      ;
2289      2DC1      . . .      ;          THIS ROUTINE WAITS UNTIL CTU NOT BUSY OR
2290      2DC1      . . .      ;          USER INTERRUPT.  DISPLAYS "CTU BUSY" MESSAG
2291      2DC1      . . .      ;          TAPES INSERTED DURING THE WAIT ARE REWOUND
2292      2DC1      . . .      ;          BEFORE BSYCHK RETURNS.
2293      2DC1      . . .      ;
2294      2DC1      . . .      ;          ENTRY:  DON'T CARE
2295      2DC1      . . .      ;
2296      2DC1      . . .      ;          EXIT :  NC => CTU NOT BUSY
2297      2DC1      . . .      ;                   C => USER INTERRUPTED
2298      2DC1      . . .      ;                   A,H,L DESTROYED
2299      2DC1      . . .      ;
2300      2DC1      . . .      ;
2301      2DC1      . . .      BSYCK0 EQU $
2302      2DC1      3A 4E FF      LDA INPDEV      ;IS INPUT A TAPE?
2303      2DC4      E6 FC .      ANI -1-LFTCTU-RGTCTU
2304      2DC6      C0 . .      RNZ              ;NO - RETURN NZ, NC
2305      2DC7      . . .      BSYCHK EQU $
2306      2DC7      CD C1 29      CALL CTMON1     ;CTU BUSY ("RUN" SET)?
2307      2DCA      CA 43 00      JZ RSTDSP       ;NO - RETURN
2308      2DCD      . . .      BSY010 EQU $
2309      2DCD      21 F8 3B      LXI H,BSYMSG   ;PRINT BUSY MESSAGE
2310      2DD0      CD A3 3C      CALL CARDIO
2311      2DD3      CD 2E 48      CALL RETSCN     ;RETURN KEY HIT?
2312      2DD6      DA 43 00      JC RSTDSP       ;YES - RETURN
2313      2DD9      CD C1 29      CALL CTMON1     ;MONITOR TAPES
2314      2DDC      C2 CD 2D      JNZ BSY010     ;STILL RUNNING - WAIT
2315      2DDF      CD 6F 29      CALL CTMON      ;STOPPED - CHECK FOR NEW TAP
2316      2DE2      C3 C7 2D      JMP BSYCHK      ;WAIT TIL NOT BUSY
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 71
2318	2DE5	.	.	. ;*****	
2319	2DE5	.	.	. ; CTU ENTRY POINTS FOR ESCAPE SEQUENCES *****	
2320	2DE5	.	.	. ; AND USER INTERFACE *****	
2321	2DE5	.	.	. ;*****	
2322	2DE5	.	.	. ;*****	
2323	2DE5	.	.	. ; CTUERR - ERROR RETURN FROM CTU DRIVERS *	
2324	2DE5	.	.	. ;*****	
2325	2DE5	.	.	. CTUERR EQU \$	
2326	2DE5	3A	63	FF LDA UNIT0 ;TURN OFF "COMMAND COMPLETED	
2327	2DE8	E6	F7	. ANI -1-CMDEXC ;BIT	
2328	2DEA	32	63	FF STA UNIT0	
2329	2DED	.	.	. CTUER1 EQU \$	
2330	2DED	CD	B3	41 CALL IOFAIL ;SET ERROR FLAG	
2331	2DF0	.	.	. ;*****	
2332	2DF0	.	.	. ; SLTPMS - GET MESSAGE FOR SELECTED TAPE UNIT *	
2333	2DF0	.	.	. ;*****	
2334	2DF0	.	.	. SLTPMS EQU \$	
2335	2DF0	22	F1	FF SHLD MSGPT1 ;STORE POINTER TO ERROR MSG	
2336	2DF3	CD	DC	2A CALL GTCTBT ;SET UP PTR TO UNIT MSG	
2337	2DF6	.	.	. SLTPM1 EQU \$;ENTRY FOR CMPARE ROUTINE	
2338	2DF6	21	BF	3B LXI H,0LTPMS ;"LEFT TAPE" MESSAGE	
2339	2DF9	C2	FF	2D JNZ CTUER2	
2340	2DFC	21	CF	3B LXI H,0RTPMS ;"RIGHT TAPE" MESSAGE	
2341	2DFF	.	.	. CTUER2 EQU \$	
2342	2DFF	22	EF	FF SHLD MSGPT2	
2343	2E02	21	BD	3B LXI H,EOPMSG ;SET SECOND HALF TO NO	
2344	2E05	22	ED	FF SHLD MSGPT3 ;MESSAGE	
2345	2E08	AF	.	. XRA A ;SET A REG = 0	
2346	2E09	37	.	. STC	
2347	2E0A	C9	.	. RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 72
2349	2E08	.	.	;	
2350	2E08	.	.	; * * * * *	
2351	2E08	.	.	;	
2352	2E08	.	.	; CT2BUF - READ A CTU RECORD	
2353	2E08	.	.	;	
2354	2E08	.	.	; ENTRY: D,E -> STATUS OF LAST BUFFER RETURNED	
2355	2E08	.	.	; (DON'T CARE FOR FIRST READ)	
2356	2E08	.	.	; EXIT : NC => SUCCESSFUL READ	
2357	2E08	.	.	; D,E -> NEW BUFFER STATUS	
2358	2E08	.	.	; C => ERROR	
2359	2E08	.	.	; IOCERR=U => USER INTERRUPTED	
2360	2E08	.	.	; IOCERR=F => FAILURE	
2361	2E08	.	.	; MSGPTX -> ERROR MSG	
2362	2E08	.	.	; KILLS ALL REGISTERS	
2363	2E08	.	.	;	
2364	2E08	.	.	LCT2BF EQU \$;PUT FLAG FOR UNIT IN B	
2365	2E08	06	01	MVI B,LFTCTU	
2366	2E0D	C3	12	JMP CT2BUF	
2367	2E10	.	.	RCT2BF EQU \$	
2368	2E10	06	02	MVI B,RGTCTU	
2369	2E12	.	.	CT2BUF EQU \$	
2370	2E12	3A	55	LDA CMND ;GET COMMAND FOR READ INIT	
2371	2E15	4F	.	MOV C,A	
2372	2E16	CD	1F	CALL CHGBUF ;LOOK AT OTHER BUFFER	
2373	2E19	1A	.	LDAX D ;IS IT READY?	
2374	2E1A	B8	.	CMP B ;(STATUS=INPUT UNIT => YES	
2375	2E1B	CA	30	JZ C2B020 ;YES - START NEXT READ & EXI	
2376	2E1E	CD	1F	CALL CHGBUF ;NO - IS FIRST BUF READY?	
2377	2E21	1A	.	LDAX D	
2378	2E22	B8	.	CMP B	
2379	2E23	CA	30	JZ C2B020 ;YES - START NEXT READ	
2380	2E26	CD	3F	CALL RDINIT ;NO - SET UP READ	
2381	2E29	D4	C1	CNC CTMON1 ;IF NO ERROR, MONITOR TAPES	
2382	2E2C	D2	12	JNC CT2BUF ;IF NO ERROR, CHECK AGAIN	
2383	2E2F	C9	.	RET	
2384	2E30	.	.	C2B020 EQU \$	
2385	2E30	1B	.	DCX D ;D,E -> TYPE	
2386	2E31	1A	.	LDAX D ;A = RECORD TYPE	
2387	2E32	13	.	INX D	
2388	2E33	21	47	LXI H,XFRLIM ;H,L -> TRANSFER LIMIT	
2389	2E36	BE	.	CMP M ;LIMIT REACHED?	
2390	2E37	2E	55	MVI L,CMND-BASE ;(GET CMND FOR INIT)	
2391	2E39	4E	.	MOV C,M	
2392	2E3A	FC	3F	CM RDINIT ;NO-ATTEMPT TO INIT NEXT REA	
2393	2E3D	B7	.	ORA A ;NC => LAST READ OK	
2394	2E3E	C9	.	RET	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 73
2396	2E3F	. . .	;	
2397	2E3F	. . .	; * * * * *	
2398	2E3F	. . .	;	
2399	2E3F	. . .	; RDINIT - START READING TAPE RECORD	
2400	2E3F	. . .	;	
2401	2E3F	. . .	; CALLED BY CT2BUF	
2402	2E3F	. . .	;	
2403	2E3F	. . .	; ENTRY: B=FLAG FOR UNIT (1=LFTCTU, 2=RGCTU)	
2404	2E3F	. . .	; C=TAPE COMMAND (CHECKED FOR RUN BIT)	
2405	2E3F	. . .	;	
2406	2E3F	. . .	; EXIT : NC => NO ERROR, READ STARTED IF CTU	
2407	2E3F	. . .	; NOT BUSY AND BUFFER AVAILABLE	
2408	2E3F	. . .	; C => ERROR (NO TAPE OR HARD ERROR OR	
2409	2E3F	. . .	; LAST FCN=RECORD OR EVD)	
2410	2E3F	. . .	; MSGPTX SET FOR ERROR MSG	
2411	2E3F	. . .	; A,H,L DESTROYED	
2412	2E3F	. . .	;	
2413	2E3F	. . .	;	
2414	2E3F	. . .	RDINIT EQU \$	
2415	2E3F	79 . .	MOV A,C ;TAPE RUNNING?	
2416	2E40	E6 01 .	ANI RUN	
2417	2E42	C0 . .	RNZ ;YES-RETURN (NC=>NO ERROR)	
2418	2E43	78 . .	MOV A,B ;SELECT UNIT	
2419	2E44	CD 89 2D	CALL SELACT	
2420	2E47	CD ED 2A	CALL CISCAN ;TAPE INSERTED?	
2421	2E4A	D4 18 2B	CNC CHKEVD ;AT EVD OR DATA RECORDED?	
2422	2E4D	D8 . .	RC ;REPORT ANY ERROR	
2423	2E4E	21 62 FF	LXI H,CNTRL0	
2424	2E51	7E . .	MOV A,M ;HARD ERROR?	
2425	2E52	E6 10 .	ANI HRDER1	
2426	2E54	CA 63 2E	JZ RDI020 ;NO - CONTINUE INITIALIZATIO	
2427	2E57	7E . .	MOV A,M ;YES - CLEAR INTERRUPT FLAG	
2428	2E58	E6 E7 .	ANI -1-HRDER1-SFTERR ;SOFT ERROR	
2429	2E5A	F6 04 .	ORI HRDERR ;SET HARD ERROR FLAG	
2430	2E5C	77 . .	MOV M,A	
2431	2E5D	21 62 3B	LXI H,HRDMSG ;ERROR MESSAGE	
2432	2E60	C3 E5 2D	JMP CTUERR ;REPORT ERROR AND EXIT	
2433	2E63	. . .	RDI020 EQU \$	
2434	2E63	CD AD 2B	CALL CHKLPM ;WIND TO LP IF NOT PAST LP	
2435	2E66	D8 . .	RC ;RETURN ON ERROR	
2436	2E67	. . .	; FALL INTO RDNEXT	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 74
=====
2438      2E67      . . .      ; * * * * *
2439      2E67      . . .      ;
2440      2E67      . . .      ;
2441      2E67      . . .      ;      RDNEXT - START READING NEXT TAPE RECORD
2442      2E67      . . .      ;
2443      2E67      . . .      ;      CALLED BY INTERRUPT ROUTINE AND RDINIT
2444      2E67      . . .      ;
2445      2E67      . . .      ;      ENTRY: UNIT SELECTED
2446      2E67      . . .      ;
2447      2E67      . . .      ;      EXIT : C,NZ => USER INTERRUPT
2448      2E67      . . .      ;              NC => NO ERROR
2449      2E67      . . .      ;              Z => READ STARTED
2450      2E67      . . .      ;              NZ => NO BUFFER AVAILABLE
2451      2E67      . . .      ;              A,H,L DESTROYED
2452      2E67      . . .      ;
2453      2E67      . . .      ;
2454      2E67      . . .      RDNEXT EQU $
2455      2E67      CD 2E 48      CALL RETSCN      ;CHECK FOR USER INTERRUPT
2456      2E6A      3C . .      INR A      ;(INSURE TNZ)
2457      2E6B      D8 . .      RC
2458      2E6C      CD 0C 3D      CALL GTIOBF      ;BUFFER AVAILABLE?
2459      2E6F      C0 . .      RNZ      ;NO - RETURN
2460      2E70      36 80 .      MVI M,200Q      ;YES - MARK IT BUSY
2461      2E72      22 31 FF      SHLD CTISPT      ;STORE STATUS POINTER
2462      2E75      97 . .      SUB A      ;CLEAR CONTROL FLAGS
2463      2E76      32 62 FF      STA CNTRL0
2464      2E79      3E 09 .      MVI A,9      ;EACH RECORD GETS (RE-TRY)
2465      2E7B      . . .      RDVERF EQU $      ;ENTRY FOR VERIFY MODE
2466      2E7B      32 2B FF      STA CTITRL
2467      2E7E      7D . .      MOV A,L      ;GET POINTER TO FIRST BYTE
2468      2E7F      CD 2B 3D      CALL GETPT1
2469      2E82      22 2F FF      SHLD CTIBPT      ;STORE POINTER
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 75
=====
2471     2E85      . . .      ;
2472     2E85      . . .      ; * * * * *
2473     2E85      . . .      ;
2474     2E85      . . .      ; RDSTRT - SET UP RECORD READ
2475     2E85      . . .      ;
2476     2E85      . . .      ; CALLED BY INTERRUPT ROUTINES VIA RDRTY
2477     2E85      . . .      ;
2478     2E85      . . .      ; ENTRY: CTISPT -> STATUS OF SELECTED BUFFER
2479     2E85      . . .      ; CTIBPT -> FIRST BYTE
2480     2E85      . . .      ;
2481     2E85      . . .      ; EXIT : NC (NO ERROR POSSIBLE)
2482     2E85      . . .      ; A,H,L DESTROYED
2483     2E85      . . .      ;
2484     2E85      . . .      ;
2485     2E85      . . .      RDSTRT EQU $
2486     2E85      21 E5 2E      LXI H,GETPRM ;SET STATUS=WAIT FOR PREAMBL
2487     2E88      22 33 FF      SHLD CTIADR
2488     2E88      3E 80 .      MVI A,128 ;SET GAP COUNT-DOWN=128 (>4"
2489     2E8D      32 2C FF      STA CTICNT
2490     2E90      21 9A 2E      LXI H,TIGCT0 ;FIRST, WAIT FOR GAP
2491     2E93      3E 03 .      MVI A,RUN+FWD ;START TAPE RUNNING AND RET
2492     2E95      CD 78 2A      CALL OUTCMD
2493     2E98      97 . .      SUB A ;NC,Z => READ STARTED
2494     2E99      C9 . .      RET
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 76
=====
2496     2E9A      . . .      ;
2497     2E9A      . . .      ; * * * * *
2498     2E9A      . . .      ;
2499     2E9A      . . .      ;      WATGAP - WAIT FOR THE BEGINNING OF A GAP
2500     2E9A      . . .      ;
2501     2E9A      . . .      ;      THIS ROUTINE INSURES THAT POSTAMBLE BYTES
2502     2E9A      . . .      ;      FROM ONE RECORD WILL NOT BE MISTAKEN FOR
2503     2E9A      . . .      ;      PREAMBLE BYTES OF THE NEXT.
2504     2E9A      . . .      ;
2505     2E9A      . . .      ;
2506     2E9A      . . .      TIGCT0 EQU $
2507     2E9A      7E . .      MOV  A,M      ;GET STATUS
2508     2E9B      E6 20 .      ANI  GAP      ;IN GAP?
2509     2E9D      3A 20 8B     LDA  IOCTDI   ;(CLEAR BYTE READY)
2510     2EA0      C8 . .      RZ           ;NO - CONTINUE WAITING
2511     2EA1      21 A8 2E     LXI  H,TIGCT1 ;YES - SET UP RECORD-READING
2512     2EA4      22 E1 FF     SHLD CTIVEC  ;INTERRUPT ROUTINE
2513     2EA7      C9 . .      RET
  
```


ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
2565	2EF5	AE	.	.	XRA M	78
2566	2EF6	.	.	.		
2567	2EF6	.	.	.		
2568	2EF6	F2	A7	2F	JP RDRTRY	
2569	2EF9	AE	.	.	XRA M	
2570	2EFA	32	2C	FF	STA CTICNT	
2571	2EFD	C3	10	2F	JMP GCT035	
2572	2F00	.	.	.	GCT020 EQU \$	
2573	2F00	36	FF	.	MVI M,3770	
2574	2F02	B7	.	.	ORA A	
2575	2F03	F2	07	2F	JP GCT030	
2576	2F06	34	.	.	INR M	
2577	2F07	.	.	.	GCT030 EQU \$	
2578	2F07	2B	.	.	DCX H	
2579	2F08	E6	7F	.	ANI 1770	
2580	2F0A	77	.	.	MOV M,A	
2581	2F0B	FE	02	.	CPI 2	
2582	2F0D	F2	A7	2F	JP RDRTRY	
2583	2F10	.	.	.	GCT035 EQU \$	
2584	2F10	21	16	2F	LXI H,GETLSB	
2585	2F13	C3	5E	2F	JMP GCT100	
2586	2F16	.	.	.	GETLSB EQU \$	
2587	2F16	2A	31	FF	LHLD CTISPT	
2588	2F19	2B	.	.	DCX H	
2589	2F1A	2B	.	.	DCX H	
2590	2F1B	CA	25	2F	JZ GCT040	
2591	2F1E	8E	.	.	CMP M	
2592	2F1F	C2	A7	2F	JNZ RDRTRY	
2593	2F22	21	2C	FF	LXI H,CTICNT	
2594	2F25	.	.	.		
2595	2F25	.	.	.	GCT040 EQU \$	
2596	2F25	B7	.	.	ORA A	
2597	2F26	CA	3C	2F	JZ GCT060	
2598	2F29	35	.	.	DCR M	
2599	2F2A	CA	A7	2F	JZ RDRTRY	
2600	2F2D	77	.	.	MOV M,A	
2601	2F2E	.	.	.	GCT050 EQU \$	
2602	2F2E	32	2C	FF	STA CTICNT	
2603	2F31	21	2A	FF	LXI H,CTICSM	
2604	2F34	86	.	.	ADD M	
2605	2F35	77	.	.	MOV M,A	
2606	2F36	21	43	2F	LXI H,GETDAT	
2607	2F39	C3	5E	2F	JMP GCT100	
2608	2F3C	.	.	.	GCT060 EQU \$	
2609	2F3C	35	.	.	DCR M	
2610	2F3D	CA	2E	2F	JZ GCT050	
2611	2F40	C3	A7	2F	JMP RDRTRY	
2612	2F43	.	.	.	GETDAT EQU \$	
2613	2F43	2A	2F	FF	LHLD CTIBPT	
2614	2F46	CA	4D	2F	JZ GCT070	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
2615	2F49	BE	.	.	CMP M ;VERIFY - SAME AS BYTE IN	79
2616	2F4A	C2	A7	2F	JNZ RDRTRY ;BUFFER? NO => ERROR	
2617	2F4D	.	.	.	GCT070 EQU \$	
2618	2F4D	77	.	.	MOV M,A ;STORE BYTE IN BUFFER	
2619	2F4E	23	.	.	INX H	
2620	2F4F	22	2F	FF	SHLD CTIBPT	
2621	2F52	21	2A	FF	LXI H,CTICSM ;UPDATE CHECKSUM	
2622	2F55	86	.	.	ADD M	
2623	2F56	77	.	.	MOV M,A	
2624	2F57	2E	2C	.	MVI L,CTICNT*256/256	
2625	2F59	35	.	.	DCR M ;DECREMENT COUNT	
2626	2F5A	C0	.	.	RNZ ;MORE BYTES - RETURN	
2627	2F5B	21	62	2F	LXI H,GETCSM ;GET CHECKSUM NEXT	
2628	2F5E	.	.	.	GCT100 EQU \$;UPDATE STATUS FOR NEXT	
2629	2F5E	22	33	FF	SHLD CTIADR ;SAVE ADDRESS OF NEXT SUBROU	
2630	2F61	C9	.	.	RET	
2631	2F62	.	.	.	GETCSM EQU \$	
2632	2F62	21	2A	FF	LXI H,CTICSM ;GET RUNNING CHECKSUM	
2633	2F65	BE	.	.	CMP M ;COMPARE WITH DATA	
2634	2F66	C2	A7	2F	JNZ RDRTRY ;NOT OK- RETRY	
2635	2F69	3A	65	FF	LDA IOFLGS ;READ OR VERIFY MODE?	
2636	2F6C	E6	80	.	ANI VERIFY	
2637	2F6E	C2	E0	2B	JNZ STPTP0 ;VERIFY - CLEAR TACH AND QUI	
2638	2F71	CD	DC	2A	CALL GTCTBT ;GET BIT FOR SELECTED UNIT	
2639	2F74	2A	31	FF	LHLD CTISPT	
2640	2F77	77	.	.	MOV M,A	
2641	2F78	2B	.	.	DCX H	
2642	2F79	7E	.	.	MOV A,M ;FILE MARK?	
2643	2F7A	B7	.	.	ORA A	
2644	2F7B	C2	8F	2F	JNZ GCT320 ;NO - START NEXT READ	
2645	2F7E	2A	2F	FF	LHLD CTIBPT ;YES-GET FILE NUMBER	
2646	2F81	2E	00	.	MVI L,0	
2647	2F83	7E	.	.	MOV A,M ;UPDATE CURRENT FILE NUMBER	
2648	2F84	3C	.	.	INR A	
2649	2F85	21	5E	FF	LXI H,FILNUM	
2650	2F88	77	.	.	MOV M,A ;SAVE IN RAM	
2651	2F89	3E	01	.	MVI A,EOF ;SET END OF FILE STATUS	
2652	2F8B	CD	72	32	CALL SETCT0	
2653	2F8E	97	.	.	SUB A ;0="FILE MARK" RECORD	
2654	2F8F	.	.	.	GCT320 EQU \$	
2655	2F8F	2E	61	.	MVI L,RELTAK*256/256	
2656	2F91	36	00	.	MVI M,0 ;CLEAR GAP-LENGTH COUNTER	
2657	2F93	2E	47	.	MVI L,XFRLIM ;TRANSFER LIMIT REACHED?	
2658	2F95	BE	.	.	CMP M	
2659	2F96	F2	E4	2B	JP STOPTP ;YES - STOP TAPE & RETURN	
2660	2F99	21	7D	28	LXI H,TID00 ;SET UP "DO NOTHING" ROUTINE	
2661	2F9C	22	E1	FF	SHLD CTIVEC ;TO COUNT GAP TACHS WHILE	
2662	2F9F	FB	.	.	EI ;STARTING NEXT READ	
2663	2FA0	CD	67	2E	CALL RDNEXT ;NO - START NEXT RECORD	
2664	2FA3	C8	.	.	RZ ;SUCCESSFUL RDINIT - RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 80
=====
2665     2FA4     C3  E4  2B          JMP  STOPTP      ;UNSUCCESSFUL - STOP TAPE
2666     2FA7     .   .   .          RDRTRY EQU $
2667     2FA7     21  2B  FF          LXI  H,CTITRL   ;CTITRL=0 => FATAL ERROR (TO
2668     2FAA     35  .   .          DCR  M          ;MANY RE-READS OF ONE REC)
2669     2FAB     21  62  FF          LXI  H,CNTRL0   ;(GET ERROR BITS)
2670     2FAE     7E  .   .          MOV  A,M
2671     2FAF     CA  C3  2F          JZ   GCT420     ;HARD ERROR - QUIT
2672     2FB2     F6  08  .          ORI  SFTERR     ;SOFT ERROR - SET BIT
2673     2FB4     77  .   .          MOV  M,A
2674     2FB5     CD  E4  2B          CALL STOPTP     ;STOP THE CTU
2675     2FB8     CD  CB  2F          CALL STRTRY     ;START THE RETRY (BACK 2 REC
2676     2FBB     .   .   .          ;              THEN FORWARD ONE)
2677     2FBB     D8  .   .          RC              ;RETURN ON ERROR
2678     2FBC     97  .   .          SUB  A          ;SET POINTER TO FIRST BYTE
2679     2FBD     32  2F  FF          STA  CTIBPT     ;OF BUFFER
2680     2FC0     C3  85  2E          JMP  RDSTRT     ;START THE RE-READ
2681     2FC3     .   .   .          GCT420 EQU $     ;FATAL ERROR
2682     2FC3     E6  F7  .          ANI  -1-SFTERR ;TURN OFF SOFT ERROR BIT
2683     2FC5     F6  10  .          ORI  HRDER1     ;SET HARD ERROR FLAG
2684     2FC7     77  .   .          MOV  M,A
2685     2FC8     C3  45  2C          JMP  FWDSP1     ;CLEAR THE BAD RECORD
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 81
2687	2FCB	.	.	.	;
2688	2FCB	.	.	.	; * * * * *
2689	2FCB	.	.	.	;
2690	2FCB	.	.	.	; STRTRY - SET UP A READ/RECORD RETRY
2691	2FCB	.	.	.	;
2692	2FCB	.	.	.	; EXIT : TAPE SPACED BACK TWO (TO GET OVER
2693	2FCB	.	.	.	; SCRAPER) AND FORWARD ONE.
2694	2FCB	.	.	.	; A,H,L DESTROYED
2695	2FCB	.	.	.	;
2696	2FCB	.	.	.	;
2697	2FCB	.	.	.	; STRTRY EQU \$
2698	2FCB	3A	65	FF	LDA IOFLGS ;VERIFY MODE?
2699	2FCE	E6	80	.	ANI VERIFY
2700	2FD0	C2	E1	2F	JNZ SRT500 ;YES - DON'T TOUCH SFTCNT
2701	2FD3	21	5D	FF	LXI H,SFTCNT ;SFTCNT<-0 => DISPLAY "RETRY
2702	2FD6	35	.	.	DCR M ;MSG (TOO MANY RETRIES ON
2703	2FD7	C2	E1	2F	JNZ SRT500 ;ONE PASS)
2704	2FDA	34	.	.	INR M ;RESET SFTCNT TO 1 FOR NEXT
2705	2FDB	21	B7	3B	LXI H,RTRYMS ;RETRY
2706	2FDE	CD	A3	3C	CALL CARDIO ;DISPLAY "RETRY"
2707	2FE1	.	.	.	SRT500 EQU \$
2708	2FE1	21	02	00	LXI H,2 ;BACKSPACE TWO RECORDS TO RU
2709	2FE4	CD	79	2C	CALL BAKSPR ;TAPE OVER SCRAPER
2710	2FE7	D4	AD	2B	CNC CHKLPM ;NO ERR - MOVE TO LP IF BEHN
2711	2FEA	C4	45	2C	CNZ FWDSP1 ;NO ERR, NO ADV TO LP - FWD
2712	2FED	C3	43	00	JMP RSTDSP ;RESTORE DISPLAY AND RETURN

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 82
=====
2714     2FF0      . . .      ;
2715     2FF0      . . .      ; * * * * *
2716     2FF0      . . .      ;
2717     2FF0      . . .      ;      BUF2CT - RECORD I/O BUFFER ON TAPE
2718     2FF0      . . .      ;
2719     2FF0      . . .      ;      ENTRY: D,E -> BUFFER STATUS
2720     2FF0      . . .      ;      MDFLG2[WBSR]=1 => WRITE/BS/READ
2721     2FF0      . . .      ;
2722     2FF0      . . .      ;      EXIT : D,E -> BUFFER STATUS
2723     2FF0      . . .      ;      NC => SUCCESS
2724     2FF0      . . .      ;      BXSTAT[XXXCTU] CLEARED
2725     2FF0      . . .      ;      W/BS/R MODE => SUCCESSFUL WRITE
2726     2FF0      . . .      ;      NOT W/BS/R => SUCCESSFUL WRITE INIT
2727     2FF0      . . .      ;      C => FAILURE
2728     2FF0      . . .      ;      A,B,C,H,L DESTROYED
2729     2FF0      . . .      ;
2730     2FF0      . . .      ;
2731     2FF0      . . .      BF2LCT EQU $      ;PUT UNIT SELECT FLAG IN B
2732     2FF0      06 01      MVI B,LFTCTU
2733     2FF2      C3 F7 2F    JMP BUF2CT
2734     2FF5      . . .      BF2RCT EQU $
2735     2FF5      06 02      MVI B,RGTCTU
2736     2FF7      . . .      BUF2CT EQU $      ;WORKING ON THIS BUFFER?
2737     2FF7      1A . .      LDAX D
2738     2FF8      A0 . .      ANA B
2739     2FF9      . . .      ;*****
2740     2FF9      . . .      ; ROM BREAK 1
2741     2FF9      C3 02 30    JMP ZBRK1C
2742     2FFC      . . .      ORG CTSTRT+4000Q
2743     3000      . . .      ZBRK1 EQU $
2744     3000      54 . .      DB VERSN      ;ROM PRESENT/VERSION FLAG
2745     3001      30 . .      DB ZBRK1/256
2746     3002      . . .      ZBRK1C EQU $
2747     3002      . . .      ;*****
2748     3002      CA 13 30    JZ B2C015
2749     3005      CD 85 30    CALL WRINIT      ;THIS BUFFER NOT STARTED -
2750     3008      D4 C1 29    CNC CTMON1      ;CHECK TAPES ON NO ERROR
2751     300B      D2 F7 2F    JNC BUF2CT      ;TRY AGAIN ON NO ERROR
2752     300E      CD 70 32    CALL SETCTW      ;WRITE ERROR - SET ERROR FLA
2753     3011      37 . .      STC      ;C => ERROR
2754     3012      C9 . .      RET
2755     3013      . . .      B2C015 EQU $      ;BUFFER FINISHED - CHECK FOR
2756     3013      3A 62 FF    LDA CNTRL0      ;SUCCESSFUL COMPLETION
2757     3016      E6 20      ANI WRTERR
2758     3018      37 . .      STC      ;(SET CARRY, IN CASE)
2759     3019      C0 . .      RNZ      ;RETURN ON ERROR
2760     301A      . . .      ;*****
2761     301A      . . .      ; SUCCESSFUL WRITE START *
2762     301A      . . .      ;*****
2763     301A      3F . .      CMC      ;CLEAR ERROR FLAG
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
2764	301B	1B	.	.	DCX D ;WHAT TYPE OF RECORD?	83
2765	301C	1A	.	.	LDAX D ;-1 => NORMAL	
2766	301D	13	.	.	INX D ;0 => EOF	
2767	301E	3D	.	.	DCR A ;1 => EVD	
2768	301F	C8	.	.	RZ ;EVD - DO NOT CHK FOR W/BS/R	
2769	3020	3A	F3	FF	LDI MDPLG2 ;WRITE/BS/READ MODE?	
2770	3023	E6	20	.	ANI WBSR	
2771	3025	C8	.	.	RZ ;NO - RETURN (NC => SUCCESS)	
2772	3026	0E	09	.	MVI C,9 ;YES - INIT ATTEMPTS COUNTER	
2773	3028	.	.	.	B2C030 EQU \$;SEE IF RECORD IS DONE	
2774	3028	CD	C1	29	CALL CTMON1 ;TAPE STILL RUNNING?	
2775	302B	D8	.	.	RC ;RETURN ON STALL OR NO TAPE	
2776	302C	C2	28	30	JNZ B2C030 ;STILL RUNNING - WAIT	
2777	302F	.	.	.	B2C040 EQU \$;WRITE FINISHED	
2778	302F	21	01	00	LXI H,1 ;BACKSPACE TO READ	
2779	3032	CD	79	2C	CALL BAKSPR	
2780	3035	3E	80	.	MVI A,VERIFY ;PUT IN VERIFY MODE	
2781	3037	CD	24	2B	CALL STIOFS	
2782	303A	62	.	.	MOV H,D ;COPY BUF STATUS POINTER	
2783	303B	6B	.	.	MOV L,E	
2784	303C	3E	02	.	MVI A,2 ;ALLOW ONE RE-TRY	
2785	303E	CD	7B	2E	CALL RDVERF ;START THE VERIFY	
2786	3041	.	.	.	B2C050 EQU \$;WAIT FOR VERIFY COMPLETION	
2787	3041	CD	C1	29	CALL CTMON1 ;TAPE STOPPED YET?	
2788	3044	C2	41	30	JNZ B2C050 ;NO - CONTINUE WAITING	
2789	3047	F5	.	.	PUSH PSW ;YES - SAVE FLAGS	
2790	3048	3E	7F	.	MVI A,-1-VERIFY	
2791	304A	CD	2A	2B	CALL CLIOFS ;TURN OFF VERIFY MODE	
2792	304D	F1	.	.	POP PSW ;RECALL FLAGS	
2793	304E	D8	.	.	RC ;RETURN ON STALL OR NO TAPE	
2794	304F	2E	62	.	MVI L,CNTRL0 ;WAS VERIFY SUCCESSFUL?	
2795	3051	7E	.	.	MOV A,M ;(HRDER1 = 0)?	
2796	3052	E6	EF	.	ANI -1-HRDER1	
2797	3054	BE	.	.	CMP M	
2798	3055	C2	5D	30	JNZ B2C100 ;NO - RE-RECORD	
2799	3058	1A	.	.	LDAX D ;RELEASE BUFFER (CLEAR BIT	
2800	3059	E6	7F	.	ANI -1-BUFBSY ;IN BUF STATUS HOLDING BUF	
2801	305B	12	.	.	STAX D ;FOR UNSPECIFIED UNIT)	
2802	305C	C9	.	.	RET	
2803	305D	.	.	.	*****	
2804	305D	.	.	.	; WRITE ERROR - TRY RE-RECORDING *	
2805	305D	.	.	.	*****	
2806	305D	.	.	.	B2C100 EQU \$	
2807	305D	0D	.	.	DCR C ;FATAL ERROR (8 RETRYS)?	
2808	305E	C2	6C	30	JNZ B2C110 ;NO - INIT RE-RECORD	
2809	3061	E6	F7	.	ANI -1-SFTERR ;YES - CLEAR SOFT ERROR BIT	
2810	3063	F6	64	.	ORI DATATR+HRDERR+WRKERR ;SET HARD ERROR	
2811	3065	77	.	.	MOV M,A ;AND "DATA RECORDED" BITS	
2812	3066	.	.	.	B2C105 EQU \$;ENTRY FOR BAD EVD READ	
2813	3066	21	08	3C	LXI H,WRFMSG ;WRITE FAIL MESSAGE	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  84
=====
2814  3069   C3 E5 2D          JMP  CTUERR      ;REPORT ERROR
2815  306C   .  .  .          ;*****
2816  306C   .  .  .          ; START RE-RECORDING *
2817  306C   .  .  .          ;*****
2818  306C   .  .  .          B2C110 EQU $
2819  306C   77 .  .          MOV  M,A        ;TURN OFF CNTRL0 [HRDER1]
2820  306D   CD CB 2F        CALL STRTRY     ;BACKSP 2, FWDSP 1 FOR RETRY
2821  3070   D4 85 30        CNC  WRINIT     ;RETRY ON NO SPACING ERROR
2822  3073   21 62 FF        LXI H,CNTRL0   ;(GET ERROR BITS)
2823  3076   7E .  .          MOV  A,M
2824  3077   DA 80 30       JC   B2C150     ;ERROR IN WRINIT - QUIT
2825  307A   F6 28 .          ORI  SFTERR+WRTERR ;SUCCESSFUL WRINIT -
2826  307C   77 .  .          MOV  M,A        ;SET SOFT ERROR BITS
2827  307D   C3 28 30       JMP  B2C030     ;WAIT FOR COMPLETION
2828  3080   .  .  .          B2C150 EQU $    ;ERROR IN WRINIT
2829  3080   F6 60 .          ORI  DATATR+WRTERR ;SET ERROR BITS
2830  3082   77 .  .          MOV  M,A
2831  3083   37 .  .          STC              ;C => ERROR
2832  3084   C9 .  .          RET
=====
  
```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 85
=====
2834      3085      . . .      ;
2835      3085      . . .      ; * * * * *
2836      3085      . . .      ;
2837      3085      . . .      ; WRINIT - START WRITING TAPE RECORD
2838      3085      . . .      ;
2839      3085      . . .      ; ENTRY: B=FLAG FOR UNIT (1=LFTCTU, 2=RGCTU)
2840      3085      . . .      ; D,E -> BUFFER STATUS
2841      3085      . . .      ;
2842      3085      . . .      ; EXIT : D,E -> BUFFER STATUS
2843      3085      . . .      ; NC => NO ERROR
2844      3085      . . .      ; C => ERROR
2845      3085      . . .      ; TAPE STOPPED
2846      3085      . . .      ; IOCERR=F
2847      3085      . . .      ; MSGPTX -> ERROR MESSAGE
2848      3085      . . .      ; A,H,L DESTROYED
2849      3085      . . .      ;
2850      3085      . . .      ;
2851      3085      . . .      WRINIT EQU $
2852      3085      3A 55 FF      LDA CMND      ;IS TAPE RUNNING?
2853      3088      E6 01 .      ANI RUN      ;(RUN=1 => YES)
2854      308A      C0 . .      RNZ          ;YES - RET (NC => NO ERROR)
2855      308B      78 . .      MOV A,B      ;NO - SELECT UNIT
2856      308C      CD 89 2D      CALL SELACT
2857      308F      CD ED 2A      CALL CISCAN  ;TAPE INSERTED?
2858      3092      D8 . .      RC          ;NO - RETURN ERROR
2859      3093      3A 63 FF      LDA UNIT0    ;TAPE WRITE PROTECTED?
2860      3096      E6 04 .      ANI FPS
2861      3098      21 27 3B      LXI H,NRCMSG ;(GET PROTECT MESSAGE)
2862      309B      C2 E5 2D      JNZ CTUERR   ;YES - REPORT ERROR
2863      309E      CD AD 2B      CALL CHKLPM  ;MOVE TAPE TO LP IF BEFORE L
2864      30A1      D8 . .      RC          ;RETURN ON ERRORS
2865      30A2      . . .      ; NO ERRORS - FALL INTO WRSTRT
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 86
2867	30A2	.	.	;	
2868	30A2	.	.	; * * * * *	
2869	30A2	.	.	;	
2870	30A2	.	.	; WRSTRT - START WRITING A RECORD ON TAPE	
2871	30A2	.	.	;	
2872	30A2	.	.	; ENTRY: D,E -> BUFFER STATUS	
2873	30A2	.	.	; UNIT SELECTED	
2874	30A2	.	.	;	
2875	30A2	.	.	; EXIT : D,E -> BUFFER STATUS	
2876	30A2	.	.	; NC => NO ERROR	
2877	30A2	.	.	; CTU RECORD IN PROGRESS	
2878	30A2	.	.	; C => ERROR	
2879	30A2	.	.	; TAPE STOPPED	
2880	30A2	.	.	; IOCERR = F	
2881	30A2	.	.	; MSGPTX -> ERROR MESSAGE(S)	
2882	30A2	.	.	; A,H,L DESTROYED	
2883	30A2	.	.	;	
2884	30A2	.	.	;	
2885	30A2	.	.	WRSTRT EQU \$	
2886	30A2	CD	30 2D	CALL EVDBSP ;BACK OVER EVD IF THERE	
2887	30A5	D5	.	PUSH D ;SAVE POINTER TO BUF STATUS	
2888	30A6	CD	2A 3D	CALL GETPTR ;GET PTR TO 1ST BYTE OF BUF	
2889	30A9	22	2F FF	SHLD CTIBPT ;SAVE IT FOR INTERRUPT ROUT	
2890	30AC	EB	.	XCHG	
2891	30AD	22	31 FF	SHLD CTISPT ;SAVE STATUS PTR	
2892	30B0	CD	DC 2A	CALL GTCTBT ;CLEAR BIT FOR THIS UNIT	
2893	30B3	2F	.	CMA ;IN BUF STATUS	
2894	30B4	A6	.	ANA M	
2895	30B5	F6	80 .	ORI BUFBSY ;& SET GENERAL BUSY BIT	
2896	30B7	77	.	MOV M,A	
2897	30B8	EB	.	XCHG ;D,E -> STATUS	
2898	30B9	1B	.	DCX D ;WHAT KIND OF RECORD?	
2899	30BA	1A	.	LDAX D ;-1=>NORM; 0=>EOF; 1=>EVD	
2900	30BB	B7	.	ORA A	
2901	30BC	CA	C9 30	JZ WREOF ;WRITE END OF FILE	
2902	30BF	F2	38 31	JP WREVD ;EVD - WRITE 11" GAP	
2903	30C2	.	.	WRNORM EQU \$	
2904	30C2	16	40 .	MVI D,DATATR ;"DATA RECORDED, NOT AT EOF"	
2905	30C4	1E	19 .	MVI E,25 ;SET GAP LENGTH FOR IRG	
2906	30C6	C3	DA 30	JMP WRS020	
2907	30C9	.	.	WREOF EQU \$	
2908	30C9	11	5E FF	LXI D,FILNUM ;INCREMENT FILE NUM ON THIS	
2909	30CC	1A	.	LDAX D ;TAPE	
2910	30CD	77	.	MOV M,A ;1ST BUF BYTE <- NEW FILE NU	
2911	30CE	3C	.	INR A	
2912	30CF	12	.	STAX D	
2913	30D0	16	41 .	MVI D,DATATR+EOF ;"DATA RECORDED, AT EOF"	
2914	30D2	1E	32 .	MVI E,50 ;SET GAP LENGTH FOR FILE MAR	
2915	30D4	CD	FC 2A	CALL CHKEOF ;AT END OF FILE?	
2916	30D7	CA	14 31	JZ WRS030 ;NO - DON'T CHECK FOR EW	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 87
=====
2917      30DA      .      .      .      WRS020 EQU $
2918      30DA      CD     02     2B      CALL  CHKEW      ;PAST EARLY WARNING?
2919      30DD      D2     14     31      JNC   WRS030     ;NO - CONTINUE
2920      30E0      D1     .      .      POP   D          ;YES -
2921      30E1      CD     E4     2B      CALL  STOPTP     ;STOP THE CTU
2922      30E4      3A     F4     FF      LDA   MDFLG1     ;DOING DATA LOGGING. . .
2923      30E7      E6     10     .      ANI   EDIT
2924      30E9      21     24     FF      LXI   H,SWPCTU   ;. . .AND SWAPPING CTU'S?
2925      30EC      A6     .      .      ANA   M
2926      30ED      C2     FD     30      JNZ   WRS025     ;YES - SWAP UNITS
2927      30F0      12     .      .      STAX  D          ;RELEASE BUFFER FOR EVD WRIT
2928      30F1      CD     0E     2B      CALL  CHKEVO     ;AT END OF DATA MARK?
2929      30F4      CC     SE     2D      CZ    EVDWAT     ;NO - RECORD EVD AND WAIT
2930      30F7      21     07     3B      LXI   H,EOTMSG   ;REPORT END OF TAPE
2931      30FA      C3     E5     2D      JMP   CTUERR
2932      30FD      .      .      .      ;
2933      30FD      .      .      .      ; DATA LOGGING - SWAP CTU'S AT END OF TAPE
2934      30FD      .      .      .      ;
2935      30FD      .      .      .      WRS025 EQU $
2936      30FD      3E     08     .      MVI   A,RECRWD   ;SET FLAG SO CTMON WILL
2937      30FF      CD     24     2B      CALL  STIOFS     ;REWIND THIS UNIT
2938      3102      2E     4D     .      MVI   L,OUTDEV   ;SAVE THIS UNIT NUMBER
2939      3104      7E     .      .      MOV   A,M        ;FOR REWIND
2940      3105      E6     03     .      ANI   LFTCTU+RGCTU
2941      3107      32     24     FF      STA   SWPCTU     ;(BORROW SWPCTU)
2942      310A      7E     .      .      MOV   A,M        ;SWAP UNITS
2943      310B      EE     03     .      XRI   LFTCTU+RGCTU
2944      310D      77     .      .      MOV   M,A
2945      310E      2E     4F     .      MVI   L,IOCERR   ;-1 => TRY PUTIO AGAIN
2946      3110      36     FF     .      MVI   M,-1
2947      3112      37     .      .      STC
2948      3113      C9     .      .      RET              ;SET ERROR FLAG
2949      3114      .      .      .      WRS030 EQU $
2950      3114      21     62     FF      LXI   H,CNTRL0
2951      3117      7E     .      .      MOV   A,M        ;GET CURRENT STATUS
2952      3118      E6     01     .      ANI   EOF        ;AT END OF FILE?
2953      311A      7B     .      .      MOV   A,E        ;(GET GAP LENGTH)
2954      311B      CA     1F     31      JZ    WRS040
2955      311E      87     .      .      ADD   A          ;YES - DOUBLE GAP LENGTH
2956      311F      .      .      .      WRS040 EQU $
2957      311F      72     .      .      MOV   M,D        ;STORE NEW STATUS
2958      3120      2E     61     .      MVI   L,RELTAK-BASE
2959      3122      F3     .      .      DI
2960      3123      96     .      .      SUB   M          ;SUBTRACT CURRENT GAP LENGTH
2961      3124      32     2C     FF      STA   CTICNT     ;DOWN-COUNTER FOR INT. ROUT.
2962      3127      21     B7     31      LXI   H,PUTPR2   ;WRITE 2ND PREAM BYTE ROUTIN
2963      312A      22     33     FF      SHLD  CTIADR
2964      312D      3E     2B     .      MVI   A,RUN+FWD+REC+GEN
2965      312F      21     9C     31      LXI   H,TIPCT0   ;SET UP TO RECORD GAP, IF NO
2966      3132      CD     78     2A      CALL  OUTCMD     ;DOING SO ALREADY
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
2967	3135	FB	.	.	EI	
2968	3136	D1	.	.	POP D	;D,E -> BUFFER STATUS ON EXI
2969	3137	C9	.	.	RET	;OUTCMD SETS NC
2970	3138	.	.	.	WREVD EQU \$;WRITE AN EVD MARK
2971	3138	CD	0E	2B	CALL CHKEVO	;ALREADY AT END OF DATA?
2972	3138	C4	E5	2D	CNZ CTUERR	;YES - REPORT ERROR
2973	313E	DA	8D	31	JC WRS070	;AND QUIT
2974	3141	CD	FC	2A	CALL CHKEOF	;AT END OF FILE?
2975	3144	C2	6D	31	JNZ WRS060	;YES - WRITE EVD
2976	3147	97	.	.	SUB A	;NO - SET UP BUF FOR EOF
2977	3148	12	.	.	STAX D	
2978	3149	1B	.	.	DCX D	;MARK LENGTH = 1
2979	314A	3C	.	.	INR A	
2980	314B	12	.	.	STAX D	
2981	314C	13	.	.	INX D	
2982	314D	13	.	.	INX D	;D,E -> STATUS FOR BUF2CT
2983	314E	C5	.	.	PUSH B	
2984	314F	CD	E4	2B	CALL STOPTP	;BUF2CT REQ'S STOPPED TAPE
2985	3152	CD	DC	2A	CALL GTCTBT	;BUF2CT REQ'S B=UNIT
2986	3155	47	.	.	MOV B,A	; (BUF2CT CKS EOF IF W/BS/R
2987	3156	1A	.	.	LDAX D	;MARK BUFFER FOR THIS UNIT
2988	3157	80	.	.	ORA B	
2989	3158	12	.	.	STAX D	
2990	3159	CD	F7	2F	CALL BUF2CT	;GO WRITE THE EOF
2991	315C	C1	.	.	POP B	
2992	315D	DA	8D	31	JC WRS070	;RETURN ON ERROR
2993	3160	.	.	.	WRS050 EQU \$;WAIT FOR EOF TO BE FINISHED
2994	3160	CD	C1	29	CALL CTMON1	;CHECK FOR REMOVED TAPES
2995	3163	DA	8D	31	JC WRS070	;RETURN ON ERROR
2996	3166	C2	60	31	JNZ WRS050	;TAPE STILL RUNNING - WAIT
2997	3169	1B	.	.	DCX D	;RESTORE BUFFER TO EVD
2998	316A	3E	01	.	MVI A,1	; (1 = EVD)
2999	316C	12	.	.	STAX D	
3000	316D	.	.	.	WRS060 EQU \$;RECORD EVD
3001	316D	13	.	.	INX D	;RELEASE BUFFER
3002	316E	1A	.	.	LDAX D	
3003	316F	E6	7F	.	ANI -1-BUFBSY	
3004	3171	12	.	.	STAX D	
3005	3172	3E	6F	.	MVI A,111	;LOW BYTE OF EVD LENGTH
3006	3174	21	61	FF	LXI H,RELTAK	
3007	3177	F3	.	.	DI	;HOLD OFF TACH INTERRUPTS
3008	3178	96	.	.	SUB M	;SUBTRACT CURRENT GAP LENGTH
3009	3179	6F	.	.	MOV L,A	;H,L <- 11" EVD + 1.76" FILE
3010	317A	26	01	.	MVI H,1	;MARK - .21" STOP DISTANCE
3011	317C	22	33	FF	SHLD CTIADR	;USE CTIADR FOR DOWN COUNTER
3012	317F	3E	2B	.	MVI A,RUN+FWD+REC+GEN	
3013	3181	21	8F	31	LXI H,TIWEVD	;SET UP INTR ROUT AND START
3014	3184	CD	78	2A	CALL OUTCMD	;IF NOT DOING SO ALREADY
3015	3187	FB	.	.	EI	
3016	3188	3E	02	.	MVI A,EVD	;MARK TAPE AT EVD

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  89
=====
3017     318A     32  62  FF          STA  CNTRL0
3018     318D     .   .   .          WRS070 EQU  $
3019     318D     D1  .   .          POP  D           ;D,E -> BUFFER STATUS
3020     318E     C9  .   .          RET              ;RETURN - OUTCMD SETS NC
3021     318F     .   .   .          ;
3022     318F     .   .   .          ; INTERRUPT ROUTINE COUNTS EVD GAP
3023     318F     .   .   .          ;
3024     318F     .   .   .          TIWEVD EQU  $
3025     318F     7E  .   .          MOV  A,M         ;TEST STATUS
3026     3190     87  .   .          ORA  A
3027     3191     F0  .   .          RP
3028     3192     2E  33  .          MVI  L,CTIADR   ;DECREMENT COUNTER
3029     3194     35  .   .          DCR  M           ;GAP FINISHED?
3030     3195     C0  .   .          RNZ
3031     3196     23  .   .          INX  H           ;H,L -> HIGH BYTE
3032     3197     35  .   .          DCR  M
3033     3198     C8  .   .          RZ
3034     3199     C3  E4  2B          JMP  STOPTP     ;YES - STOP THE TAPE
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  90
=====
3036     319C      . . .      ;
3037     319C      . . .      ; * * * * *
3038     319C      . . .      ;
3039     319C      . . .      ;      PUTGAP - RECORD A GAP ON TAPE
3040     319C      . . .      ;
3041     319C      . . .      ;      THE FOLLOWING INTERRUPT ROUTINE OUTPUTS A
3042     319C      . . .      ;      GAP OF CTICNT TACH EDGES
3043     319C      . . .      ;
3044     319C      . . .      ;
3045     319C      . . .      TIPCT0 EQU $
3046     319C      3E 00      MVI A,0      ;CLEAR BYTE READY, AND PREPA
3047     319E      32 20 8B   STA IOCTD0   ;FOR 1ST PREAMBLE BYTE
3048     31A1      B6 . .     ORA M        ;CHECK TACH
3049     31A2      F0 . .     RP          ;NO TACH - RETURN
3050     31A3      2E 2C      MVI L,CTICNT ;DECREMENT TACH COUNTER
3051     31A5      35 . .     DCR M        ;=0?
3052     31A6      C0 . .     RNZ         ;NO - KEEP GENERATING GAP
3053     31A7      21 AF 31   LXI H,TIPCT1 ;YES - GET CTU-WRITING ROUTI
3054     31AA      3E 0B      MVI A,RUN+FWD+REC ;TURN OFF GAP
3055     31AC      C3 78 2A   JMP OUTCMD
=====
  
```


ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
3107	31E0	2B	.	.	DCX H	
3108	31EE	2B	.	.	DCX H ;H,L -> LENGTH LSB	
3109	31EF	3A	2A	FF	LDA CTICSM ;ACCUMULATE CHECKSUM	
3110	31F2	86	.	.	ADD M	
3111	31F3	32	2A	FF	STA CTICSM	
3112	31F6	7E	.	.	MOV A,M ;GET LENGTH LSB	
3113	31F7	32	2C	FF	STA CTICNT ;START BYTE COUNTER	
3114	31FA	21	00	32	LXI H,PUTDAT ;PUT DATA NEXT	
3115	31FD	C3	19	32	JMP PCT100	
3116	3200	.	.	.	PUTDAT EQU \$;WRITE A DATA BYTE	
3117	3200	2A	2F	FF	LHLD CTIBPT ;GET POINTER INTO BUFFER	
3118	3203	3A	2A	FF	LDA CTICSM ;ACCUMULATE CHECKSUM	
3119	3206	86	.	.	ADD M	
3120	3207	32	2A	FF	STA CTICSM	
3121	320A	7E	.	.	MOV A,M ;GET BYTE	
3122	320B	23	.	.	INX H ;POINT TO NEXT BYTE	
3123	320C	22	2F	FF	SHLD CTIBPT ;SAVE POINTER	
3124	320F	21	2C	FF	LXI H,CTICNT ;COUNT DOWN	
3125	3212	35	.	.	DCR M ;OUT OF BYTES YET?	
3126	3213	C2	1C	32	JNZ PCT200 ;NO	
3127	3216	21	20	32	LXI H,PUTCSM ;YES - WRITE CHECKSUM NEXT	
3128	3219	.	.	.	PCT100 EQU \$;UPDATE POINTER TO SERVICE R	
3129	3219	22	33	FF	SHLD CTIADR	
3130	321C	.	.	.	PCT200 EQU \$;OUTPUT BYTE	
3131	321C	32	20	8B	STA IOCTDO	
3132	321F	C9	.	.	RET	
3133	3220	.	.	.	PUTCSM EQU \$;WRITE CHECKSUM	
3134	3220	3A	2A	FF	LDA CTICSM ;GET CHECKSUM	
3135	3223	21	29	32	LXI H,PUTP01 ;PUT 1ST POSTAMBLE BYTE NEXT	
3136	3226	C3	19	32	JMP PCT100	
3137	3229	.	.	.	PUTP01 EQU \$;FIRST POSTAMBLE BYTE = 1	
3138	3229	3E	05	.	MVI A,5 ;SET UP REST OF POSTAMBLE -	
3139	322B	32	2C	FF	STA CTICNT ;3 BYTES "0" AND 2 TO MAKE	
3140	322E	21	36	32	LXI H,PUTPOS ;SURE THEY ARE OUT	
3141	3231	3E	01	.	MVI A,1 ;GET FIRST BYTE	
3142	3233	C3	19	32	JMP PCT100	
3143	3236	.	.	.	PUTPOS EQU \$;POSTAMBLE	
3144	3236	97	.	.	SUB A ;OUTPUT A "0"	
3145	3237	32	20	8B	STA IOCTDO	
3146	323A	21	2C	FF	LXI H,CTICNT ;ALL FIVE BYTES OUT?	
3147	323D	35	.	.	DCR M	
3148	323E	C0	.	.	RNZ ;NO - WAIT FOR MORE	
3149	323F	32	61	FF	STA RELTAK ;CLEAR TACH-GAP COUNTER	
3150	3242	21	7D	28	LXI H,TID00 ;SET UP "DO NOTHING" ROUTINE	
3151	3245	3E	28	.	MVI A,RUN+FWD+REC+GEN	
3152	3247	CD	78	2A	CALL OUTCMD ;START RECORDING GAP	
3153	324A	3A	00	8B	LDA IOCTSI ;CLEAR ANY PENDING INTERRUPT	
3154	324D	FB	.	.	EI ;ENABLE INTERRUPTS	
3155	324E	3A	F3	FF	LDA MDFLG2 ;WRITE/BS/READ MODE?	
3156	3251	E6	20	.	ANI WBSR	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
3157	3253	C2	E4	2B	JNZ STOPTP ;IF SO, QUIT	93
3158	3256	D5	.	.	PUSH D ;SAVE D AND E	
3159	3257	2A	31	FF	LHLD CTISPT ;FREE THE BUFFER	
3160	325A	7E	.	.	MOV A,M	
3161	325B	E6	7F	.	ANI -1-BUFBSY	
3162	325D	77	.	.	MOV M,A	
3163	325E	EB	.	.	XCHG ;D,E -> STATUS	
3164	325F	CD	DC	2A	CALL GTCTBT ;GET BIT FOR THIS CTU	
3165	3262	6F	.	.	MOV L,A ;SAVE TO TEST OTHER BUF	
3166	3263	CD	1F	3D	CALL CHGBUF ;D,E -> STATUS FOR OPP BUF	
3167	3266	1A	.	.	LDAX D ;UNIT BIT SET => BUF READY	
3168	3267	A5	.	.	ANA L ;FOR THIS UNIT TO RECORD	
3169	3268	CC	E4	2B	CZ STOPTP ;BUF NOT READY - STOP TAPE	
3170	326B	C4	A2	30	CNZ WRSTRT ;BUF READY - START WRITING	
3171	326E	D1	.	.	POP D ;RESTORE REGISTERS	
3172	326F	D0	.	.	RNC ;NO ERRORS - RETURN	
3173	3270	.	.	.	SETCTW EQU \$;ERROR - SET WRITE ERROR FLA	
3174	3270	3E	20	.	MVI A,WRTErr	
3175	3272	.	.	.	*****	
3176	3272	.	.	.	; SETCTO - SET FLAG IN "CNTRLO" *	
3177	3272	.	.	.	*****	
3178	3272	.	.	.	;	
3179	3272	.	.	.	; ENTRY A = FLAG TO BE SET	
3180	3272	.	.	.	;	
3181	3272	.	.	.	; EXIT A = NEW VALUE OF "CNTRLO"	
3182	3272	.	.	.	; H,L = CNTRLO	
3183	3272	.	.	.	;	
3184	3272	.	.	.	SETCTO EQU \$	
3185	3272	21	62	FF	LXI H,CNTRLO	
3186	3275	B6	.	.	ORA M ;ADD IN FLAG	
3187	3276	77	.	.	MOV M,A ;UPDATE FLAG VALUE	
3188	3277	C9	.	.	RET ;RETURN	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
3190	3278	. . .	;	94
3191	3278	. . .	; * * * * *	
3192	3278	. . .	;	
3193	3278	. . .	; EVDRED - READ BEYOND EVD	
3194	3278	. . .	;	
3195	3278	. . .	; ENTRY: DON'T CARE	
3196	3278	. . .	;	
3197	3278	. . .	; EXIT : TAPE POSITIONED JUST BEYOND FIRST	
3198	3278	. . .	; DATA BLOCK AFTER EVD GAP	
3199	3278	. . .	;	
3200	3278	. . .	; COMMAND IGNORED IF NOT AT EVD OR RECORDING	
3201	3278	. . .	; OR IF INPUT DEVICE IS NOT CTU.	
3202	3278	. . .	;	
3203	3278	. . .	;	
3204	3278	. . .	; EVDRED EQU \$	
3205	3278	CD C1 2D	CALL BSYCKO ;IS TAPE SELECTED AND FREE?	
3206	3278	D8 . .	RC ;RETURN ON USER INTERRUPT	
3207	3278	C0 . .	RNZ ;RETURN ON NOT SELECTED	
3208	3278	3A 4E FF	LDA INPDEV ;(GET INPUT DEV FOR SELECT	
3209	3280	CD 89 2D	CALL SELECT ;YES - SELECT UNIT	
3210	3283	CD 52 2D	CALL REVEVD ;WRITE EVD IF RECORDING	
3211	3286	CD 0E 2B	CALL CHKEVO ;AT EVD ALREADY?	
3212	3289	C8 . .	RZ ;NO - RETURN	
3213	328A	CD 02 2B	CALL CHKEW ;PAST EARLY WARNING?	
3214	328D	D8 . .	RC ;YES - QUIT	
3215	328E	21 AB 32	LXI H,TIEDRO ;SET UP INTERRUPT ROUTINE	
3216	3291	3E 03 .	MVI A,RUN+FWD ;RUN FORWARD	
3217	3293	CD 78 2A	CALL OUTCMD	
3218	3296	. . .	; EDR050 EQU \$	
3219	3296	CD 2E 48	CALL RETSCN ;USER INTERRUPT?	
3220	3299	DA E4 2B	JC STOPTP ;YES - STOP TAPE AND QUIT	
3221	329C	CD C1 29	CALL CTMON1 ;IF NOT, TAPE ERROR?	
3222	329F	C2 96 32	JNZ EDR050 ;TAPE STILL RUNNING - WAIT	
3223	32A2	D8 . .	RC ;RETURN IF ANY ERRORS	
3224	32A3	3E FD .	MVI A,-1-EVD ;NO ERRORS - CLEAR EVD FLAG	
3225	32A5	. . .	;*****	
3226	32A5	. . .	; CLRCT0 - CLEAR FLAG IN "CNTRL0" *	
3227	32A5	. . .	;*****	
3228	32A5	. . .	;	
3229	32A5	. . .	; ENTRY A = -1-(FLAG TO BE CLEARED)	
3230	32A5	. . .	;	
3231	32A5	. . .	; EXIT A = NEW VALUE FOR "CNTRL0"	
3232	32A5	. . .	; H,L = CNTRL0	
3233	32A5	. . .	;	
3234	32A5	. . .	; CLRCT0 EQU \$	
3235	32A5	21 62 FF	LXI H,CNTRL0	
3236	32A8	A6 . .	ANA M ;CLEAR THE FLAG BIT	
3237	32A9	77 . .	MOV M,A ;STORE NEW VALUE	
3238	32AA	C9 . .	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 95
=====
3240      32AB      . . .      ;*****
3241      32AB      . . .      ; INTERRUPT ROUTINE - RUN TO END OF GAP *
3242      32AB      . . .      ;*****
3243      32AB      . . .      TIEDRO EQU $
3244      32AB      7E . . .      MOV A,M      ;GET CTSTAT
3245      32AC      E6 20 . . .      ANI GAP      ;END OF GAP YET?
3246      32AE      C0 . . .      RNZ          ;NO - CONTINUE WAITING
3247      32AF      21 86 32 . . .      LXI H,TIEDR1 ;YES - SET UP ROUTINE TO
3248      32B2      22 E1 FF . . .      SHLD CTIVEC  ;WAIT FOR GAP
3249      32B5      C9 . . .      RET
3250      32B6      . . .      ;*****
3251      32B6      . . .      ; INTERRUPT ROUTINE - RUN TO START OF NEXT GAP *
3252      32B6      . . .      ;*****
3253      32B6      . . .      TIEDR1 EQU $
3254      32B6      7E . . .      MOV A,M      ;GET CTSTAT
3255      32B7      E6 20 . . .      ANI GAP      ;START OF NEXT GAP YET?
3256      32B9      C8 . . .      RZ          ;NO - CONTINUE WAITING
3257      32BA      C3 E0 2B . . .      JMP STPTP0   ;YES - QUIT
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 96
=====
3259     32BD      . . .      ;
3260     32BD      . . .      ; * * * * *
3261     32BD      . . .      ;
3262     32BD      . . .      ; USRCMR - COMPARE ONE RECORD
3263     32BD      . . .      ; USRCMF - COMPARE ONE FILE
3264     32BD      . . .      ; USRCMA - COMPARE ALL DATA
3265     32BD      . . .      ; (USER-INITIATED VERSIONS)
3266     32BD      . . .      ;
3267     32BD      . . .      ; ENTRY: INPDEV, OUTDEV EACH INDICATE ONE
3268     32BD      . . .      ;         DISTINCT DEVICE
3269     32BD      . . .      ;
3270     32BD      . . .      ; EXIT : ALL REGISTERS DESTROYED
3271     32BD      . . .      ;
3272     32BD      . . .      ;
3273     32BD      . . .      USRCMR EQU $           ;COMPARE ALL
3274     32BD      05 . .      DCR B                 ;TRANSFER LIMIT = -1
3275     32BE      . . .      USRCMF EQU $         ;COMPARE FILE
3276     32BE      05 . .      DCR B                 ;TRANSFER LIMIT = 0
3277     32BF      . . .      USRCMA EQU $        ;COMPARE ALL (LIMIT = 1)
3278     32BF      37 . .      STC                  ;C => COMPARE
3279     32C0      C3 CA 3E      JMP XFRD2D
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 97
=====
3281     32C3      . . .      ;
3282     32C3      . . .      ; * * * * *
3283     32C3      . . .      ;
3284     32C3      . . .      ;      CMPBFS - COMPARE I/O BUFFERS
3285     32C3      . . .      ;
3286     32C3      . . .      ;      ENTRY:  B = FILE # (USED FOR ERROR REPORT)
3287     32C3      . . .      ;              C = REC # (USED FOR ERROR REPORT)
3288     32C3      . . .      ;              CMPLIM = LIMIT OF COMPARE
3289     32C3      . . .      ;              IOBUF1 AND IOBUF2 CONTAIN RECORDS
3290     32C3      . . .      ;              TO BE COMPARED
3291     32C3      . . .      ;
3292     32C3      . . .      ;      EXIT :  C => ERROR
3293     32C3      . . .      ;              MESSAGE SET UP IN MSGPTX
3294     32C3      . . .      ;              A-L DESTROYED
3295     32C3      . . .      ;              NC => RECORDS IDENTICAL
3296     32C3      . . .      ;              A = RECORD TYPE
3297     32C3      . . .      ;              A,D-L DESTROYED
3298     32C3      . . .      ;
3299     32C3      . . .      ;
3300     32C3      . . .      ;      CMPBFS EQU $
3301     32C3      21 39 FF      LXI H,B1TYPE ;H,L -> TYPE OF FIRST BUFFER
3302     32C6      11 36 FF      LXI D,B2TYPE ;D,E -> TYEP OF SECOND BUFFE
3303     32C9      1A . .      LDAX D      ;COMPARE BUFFER TYPES
3304     32CA      BE . .      CMP M
3305     32CB      CA F3 32      JZ CPB100   ;SAME - COMPARE CONTENTS
3306     32CE      F2 D2 32      JP CPB010   ;BUF 2 GREATER
3307     32D1      EB . .      XCHG      ;BUF 1 GREATER
3308     32D2      . . .      CPB010 EQU $ ;DIFF TYPES; D,E->GREATER
3309     32D2      1A . .      LDAX D      ;GET LARGER TYPE NUMBER
3310     32D3      B7 . .      ORA A      ;WHAT TYPE OF BOUNDARY?
3311     32D4      21 15 3C      LXI H,EOFMSG ;END OF FILE MESSAGE
3312     32D7      CA DD 32      JZ CPB040   ;EOF -
3313     32DA      21 1A 3B      LXI H,EVDMSG ;EVD -
3314     32DD      . . .      CPB040 EQU $
3315     32DD      22 F1 FF      SHLD MSGPT1 ;STORE ERROR TYPE
3316     32E0      13 . .      INX D      ;D,E -> BUFFER STATUS
3317     32E1      1A . .      LDAX D      ;GET UNIT CAUSING ERROR
3318     32E2      E6 01 .      ANI LFTCTU ;WAS IT THE LEFT CTU?
3319     32E4      CD F6 2D      CALL SLTPM1 ;SET UP MESSAGE
3320     32E7      1A . .      LDAX D      ;IS ERROR UNIT A CTU?
3321     32E8      E6 03 .      ANI LFTCTU+RGCTU
3322     32EA      37 . .      STC      ;(FLAG ERROR)
3323     32EB      C0 . .      RNZ      ;YES - QUIT
3324     32EC      21 1A 92      LXI H,ZMSGAL ;NO - MUST BE ALTERNATE I/O
3325     32EF      22 EF FF      SHLD MSGPT2
3326     32F2      C9 . .      RET
3327     32F3      . . .      CPB100 EQU $ ;RECORDS SAME TYPE
3328     32F3      B7 . .      ORA A      ;WHAT KIND OF RECORDS?
3329     32F4      F0 . .      RP      ;RETURN IF NOT DATA RECORDS
3330     32F5      . . .      CPB300 EQU $ ;DATA RECORDS -
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 98
=====
3331     32F5     3A  38  FF          LDA  B1LEN          ;COMPARE LENGTHS
3332     32F8     21  35  FF          LXI  H,B2LEN
3333     32FB     BE  .   .          CMP  M              ;ARE THEY THE SAME?
3334     32FC     CA  42  33          JZ   CPB350         ;YES - CHECK CONTENTS
3335     32FF     21  22  3C          LXI  H,DLRMSG      ;NO - REPORT ERROR
3336     3302     22  F1  FF          SHLD MSGPT1        ;"DIFFERENT LENGTH RECORDS"
3337     3305     21  14  3C          LXI  H,NULMSG      ;NO MESSAGE - ALLOWS FOLLOWN
3338     3308     22  EF  FF          SHLD MSGPT2        ;TO HANDLE DIFF BYTES TOO
3339     330B     .   .   .          CPB310 EQU $         ;DECIDE WHETHER TO PRINT FIL
3340     330B     21  BD  3B          LXI  H,EOPMSG      ;AND/OR RECORD NUMBERS
3341     330E     22  ED  FF          SHLD MSGPT3        ;PUT AN END OF MSG IN EVERY
3342     3311     22  E9  FF          SHLD MSGPT5        ;POSSIBLE LOCATION
3343     3314     22  E5  FF          SHLD MSGPT7
3344     3317     21  3D  FF          LXI  H,B2DBUF      ;H,L -> BUF POSITION FOR NEX
3345     331A     .   .   .          ;                               DIGITAL OUTPUT
3346     331A     3A  46  FF          LDA  CMPLIM        ;WHAT IS THE COMPARE LIMIT?
3347     331D     B7  .   .          ORA  A
3348     331E     FA  40  33          JM   CPB330         ;ONE RECORD - PRINT NO MORE
3349     3321     CA  33  33          JZ   CPB320         ;FILE - PRINT RECORD NUMBER
3350     3324     22  E7  FF          SHLD MSGPT6        ;ALL - PRINT FILE NUMBER, TO
3351     3327     78  .   .          MOV  A,B           ;A <- FILE NUMBER
3352     3328     CD  98  3C          CALL DSPNUM
3353     332B     EB  .   .          XCHG                ;SAVE DIGIT PTR
3354     332C     21  58  3C          LXI  H,FILMSG      ;", FILE"
3355     332F     22  E9  FF          SHLD MSGPT5
3356     3332     EB  .   .          XCHG                ;GET DIGIT PTR BACK
3357     3333     .   .   .          CPB320 EQU $         ;REPORT RECORD NUMBER
3358     3333     22  EB  FF          SHLD MSGPT4        ;POINTER TO DIGITS
3359     3336     79  .   .          MOV  A,C           ;A <- RECORD NUMBER
3360     3337     CD  98  3C          CALL DSPNUM
3361     333A     21  51  3C          LXI  H,RECMMSG     ;", RECORD"
3362     333D     22  ED  FF          SHLD MSGPT3
3363     3340     .   .   .          CPB330 EQU $         ;RETURN
3364     3340     37  .   .          STC
3365     3341     C9  .   .          RET
3366     3342     .   .   .          CPB350 EQU $         ;LENGTHS SAME, COMPARE CONTN
3367     3342     21  00  FC          LXI  H,IOBUF1      ;GET POINTERS TO BUFFERS
3368     3345     11  00  FD          LXI  D,IOBUF2
3369     3348     C5  .   .          PUSH B             ;SAVE B,C
3370     3349     3A  38  FF          LDA  B1LEN
3371     334C     4F  .   .          MOV  C,A           ;C-REG USED AS DOWN-COUNTER
3372     334D     .   .   .          CPB360 EQU $         ;COMPARE ONE BYTE
3373     334D     1A  .   .          LDAX D
3374     334E     BE  .   .          CMP  M
3375     334F     23  .   .          INX  H             ;UPDATE POINTERS
3376     3350     13  .   .          INX  D
3377     3351     C2  5C  33          JNZ  CPB370        ;DIFFERENT - REPORT ERROR
3378     3354     0D  .   .          DCR  C             ;FINISHED?
3379     3355     C2  4D  33          JNZ  CPB360        ;NO - TEST NEXT BYTE
3380     3358     C1  .   .          POP  B             ;IDENTICAL DATA RECORDS
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 99
=====
3381      3359      3E FF .      MVI A,-1      ; -1 => DATA RECORDS
3382      335B      C9 . .      RET
3383      335C      . . .      CPB370 EQU $      ; REPORT DIFFERENT BYTES
3384      335C      21 3C 3C      LXI H,DIFMSG  ; "DIFFERENCE IN BYTE"
3385      335F      22 F1 FF      SHLD MSGPT1
3386      3362      21 9A FE      LXI H,DSPSTR+75
3387      3365      22 EF FF      SHLD MSGPT2  ; POINT TO BYTE NUM DIGITS
3388      3368      7B . .      MOV A,E      ; LOW BYTE OF BUF PTR IS BYT
3389      3369      CD 98 3C      CALL DSPNUM  ; CONVERT BYTE NUM TO DECIMAL
3390      336C      C1 . .      POP B      ; RECALL FILE AND RECORD NUMS
3391      336D      C3 0B 33      JMP CPB310  ; REPORT FILE AND/OR RECORD #
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 100
3393	3370	.	.	*****	
3394	3370	.	.	; ACCUMULATE UNIT STATUS *	
3395	3370	.	.	*****	
3396	3370	.	.	STLCT EQU \$;GET STATUS FOR LEFT CTU	
3397	3370	06	01	MVI B,LFTCTU ;BIT FOR LEFT TAPE	
3398	3372	3E	10	MVI A,USL ;LEFT TAPE SELECT	
3399	3374	C3	7A 33	JMP STCT	
3400	3377	.	.	STRCT EQU \$;GET STATUS FOR RIGHT CTU	
3401	3377	06	02	MVI B,RGTCTU	
3402	3379	97	.	SUB A	
3403	337A	.	.	STCT EQU \$	
3404	337A	21	55 FF	LXI H,CMND ;IS THIS UNIT SELECTED?	
3405	337D	AE	.	XRA M ;(CHECK SELECT OF LAST CMD	
3406	337E	E6	10	ANI USL	
3407	3380	7E	.	MOV A,M ;(GET COMMAND)	
3408	3381	21	63 FF	LXI H,UNITO ;(AND POINTER TO STATUS)	
3409	3384	CA	8A 33	JZ STC010 ;YES -	
3410	3387	2E	5C	MVI L,SFTCNT-1*256/256 ;NO - GET PTR TO	
3411	3389	.	.	; OTHER STATUS WORDS	
3412	3389	97	.	SUB A	
3413	338A	.	.	STC010 EQU \$	
3414	338A	E6	01	ANI 1 ;C<-1 => BUSY (UNIT SELECTED	
3415	338C	4F	.	MOV C,A ;RUNNING)	
3416	338D	56	.	MOV D,M ;D <- UNITO	
3417	338E	2B	.	DCX H	
3418	338F	5E	.	MOV E,M ;E <- CNTRL0	
3419	3390	.	.	*****	
3420	3390	.	.	; B = 1 (LEFT CTU) OR 2 (RIGHT CTU) *	
3421	3390	.	.	; *	
3422	3390	.	.	; C = 1 => BUSY; C = 0 => NOT BUSY *	
3423	3390	.	.	; *	
3424	3390	.	.	; D = UNITO () => NOT USED FOR STATUS *	
3425	3390	.	.	; 1 - LPM 10 - CMDEXC 100 - (LP) *	
3426	3390	.	.	; 2 - (LSTFWD) 20 - ---- 200 - EW *	
3427	3390	.	.	; 4 - FPS 40 - (BOT) *	
3428	3390	.	.	; *	
3429	3390	.	.	; E = CNTRL0 *	
3430	3390	.	.	; 1 - EOF 10 - SFTERR 100 - DATATR *	
3431	3390	.	.	; 2 - EVD 20 - (HRDER1) 200 - ---- *	
3432	3390	.	.	; 4 - HRDERR 40 - WRTERR *	
3433	3390	.	.	*****	
3434	3390	2E	4B	MVI L,I0STA3	
3435	3392	.	.	*****	
3436	3392	.	.	; ACCUMULATE 3RD STATUS BYTE *	
3437	3392	.	.	*****	
3438	3392	3A	66 FF	LDA CTSTAT	
3439	3395	A0	.	ANA B ;TAPE INSERTED?	
3440	3396	C2	A0 33	JNZ STC020	
3441	3399	77	.	MOV M,A ;NO - STAT3 = 0	
3442	339A	2B	.	DCX H	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 101
3443	339B	36	01	.	MVI M,1 ;STAT2 = 1 (BUSY)	
3444	339D	2B	.	.	DCX H	
3445	339E	77	.	.	MOV M,A ;STAT1 = 0	
3446	339F	C9	.	.	RET	
3447	33A0	.	.	.	STC020 EQU \$;YES -	
3448	33A0	3E	01	.	MVI A,1 ;SET "TAPE INSERTED" = 1	
3449	33A2	B3	.	.	ORA E	
3450	33A3	E6	0F	.	ANI HRDERR+SFTERR+EVD+1	
3451	33A5	77	.	.	MOV M,A	
3452	33A6	2B	.	.	DCX H	
3453	33A7	.	.	.	*****	
3454	33A7	.	.	.	; ACCUMULATE 2ND STATUS BYTE *	
3455	33A7	.	.	.	*****	
3456	33A7	7A	.	.	MOV A,D	
3457	33A8	E6	0C	.	ANI FPS+CMDEXC	
3458	33AA	B1	.	.	ORA C ;ADD BUSY BIT	
3459	33AB	4F	.	.	MOV C,A ;TEMPORARY STORAGE	
3460	33AC	7B	.	.	MOV A,E ;ANY ERRORS?	
3461	33AD	E6	0C	.	ANI HRDERR+SFTERR	
3462	33AF	CA	BB	33	JZ STC030 ;NO -	
3463	33B2	7B	.	.	MOV A,E ;YES - READ OR WRITE?	
3464	33B3	2F	.	.	CMA	
3465	33B4	E6	20	.	ANI WRTERR	
3466	33B6	CA	BB	33	JZ STC030 ;WRITE ERROR	
3467	33B9	3E	02	.	MVI A,2 ;READ ERROR, SET BIT	
3468	33BB	.	.	.	STC030 EQU \$	
3469	33BB	B1	.	.	ORA C	
3470	33BC	77	.	.	MOV M,A	
3471	33BD	2B	.	.	DCX H	
3472	33BE	.	.	.	*****	
3473	33BE	.	.	.	; ACCUMULATE 1ST STATUS BYTE *	
3474	33BE	.	.	.	*****	
3475	33BE	7A	.	.	MOV A,D	
3476	33BF	E6	81	.	ANI LPM+EW	
3477	33C1	07	.	.	RLC	
3478	33C2	07	.	.	RLC	
3479	33C3	4F	.	.	MOV C,A ;TEMPORARY STORAGE	
3480	33C4	7B	.	.	MOV A,E	
3481	33C5	E6	A1	.	ANI EOF+WRTERR+EOFINH	
3482	33C7	F2	CC	33	JP STC040 ;INHIBIT REPORTING EOF?	
3483	33CA	E6	20	.	ANI WRTERR ;YES - TURN IT OFF	
3484	33CC	.	.	.	STC040 EQU \$	
3485	33CC	07	.	.	RLC	
3486	33CD	07	.	.	RLC	
3487	33CE	07	.	.	RLC	
3488	33CF	B1	.	.	ORA C	
3489	33D0	77	.	.	MOV M,A	
3490	33D1	C9	.	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 102
3492	33D2	.	.	. ;*****	
3493	33D2	.	.	. ; CONTROL FUNCTION ENTRY TABLE *	
3494	33D2	.	.	. ;*****	
3495	33D2	.	.	. CTCTLT EQU \$	
3496	33D2	66	2B	. DW RWDBOT ;REWIND TO BEGINNING OF TAPE	
3497	33D4	07	34	. DW SPCPRC ;SPACE OVER P RECORDS	
3498	33D6	6A	34	. DW SPCPFL ;SPACE OVER FILES OR TO #P	
3499	33D8	D0	35	. DW SREVD ;LOCATE EVD MARK	
3500	33DA	CC	28	. DW ENDBAK ;CONDITION TAPE	
3501	33DC	3D	2D	. DW EOFC ;RECORD FILE MARK	
3502	33DE	69	2D	. DW EVDC ;RECORD EVD MARK	
3503	33E0	F7	35	. DW TSTCTU ;RELIABILITY TEST FOR CTU	
3504	33E2	0C	34	. DW REMSPC ;SPACE WITH NO EVD RECORDING	
3505	33E4	D4	36	. DW STWBSR ;ENTER WRITE/BAKSPC/READ MOD	
3506	33E6	DC	36	. DW CLWBSR ;EXIT WRITE/BAKSPC/READ MODE	
3507	33E8	.	.	. ;*****	
3508	33E8	.	.	. ; PERFORM CONTROL FUNCTIONS ON CTU *	
3509	33E8	.	.	. ;*****	
3510	33E8	.	.	. CTLLCT EQU \$	
3511	33E8	CD	C7	2D CALL BSYCHK ;CHECK WHETHER CTU BUSY	
3512	33EB	D8	.	. RC ;RETURN ON USER INTERRUPT	
3513	33EC	CD	8D	2D CALL SELLCT ;SELECT LEFT CTU	
3514	33EF	.	.	. CTLCT EQU \$	
3515	33EF	3E	08	. MVI A,CMDEXC ;SET "COMMAND EXECUTED" BIT	
3516	33F1	CD	9B	28 CALL STUNTO	
3517	33F4	3A	D8	FF LDA IOCTYP ;GET CONTROL CODE	
3518	33F7	21	D2	33 LXI H,CTCTLT ;GET TABLE BASE ADDRESS	
3519	33FA	C3	9A	41 JMP INDJMP ;PERFORM FUNCTION	
3520	33FD	.	.	. CTRLCT EQU \$	
3521	33FD	CD	C7	2D CALL BSYCHK ;SEE WHETHER CTU IS BUSY	
3522	3400	D8	.	. RC ;RETURN ON USER INTERRUPT	
3523	3401	CD	BA	2D CALL SELRCT ;SELECT RIGHT CTU	
3524	3404	C3	EF	33 JMP CTLCT	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 103
3526	3407	.	.	. ;*****	
3527	3407	.	.	. ; SPCPRC - SPACE P RECORDS *	
3528	3407	.	.	. ;*****	
3529	3407	.	.	. SPCPRC EQU \$	
3530	3407	06	01	. MVI B,1 ;SET TO INCREMENT IF NEED EVD	
3531	3409	CD	52	2D CALL REVEVD ;RECORD EVD IF NEEDED	
3532	340C	.	.	. REMSPC EQU \$;ESC SEQ SKIP W/O EVD	
3533	340C	3E	03	. MVI A,EOF+EVD ;CLEAR ALL BUT EOF AND EVD	
3534	340E	CD	A5	32 CALL CLRCT0 ;CLEAR EOF INHIBIT	
3535	3411	CD	ED	2A CALL CISCAN ;TAPE INSERTED?	
3536	3414	D8	.	. RC ;NO - RETURN ERROR	
3537	3415	3A	DC	FF LDA IOPSGN ;WHICH DIRECTION?	
3538	3418	87	.	. ADD A	
3539	3419	2A	D5	FF LHLD IOCCNT ;(GET NUMBER OF RECORDS)	
3540	341C	FA	25	34 JM SPC002	
3541	341F	CD	57	2C CALL FWDSPX ;FORWARD - START MOVEMENT	
3542	3422	C3	28	34 JMP SPC005	
3543	3425	.	.	. SPC002 EQU \$	
3544	3425	CD	81	2C CALL BAKSPX ;BACKWARDS - START MOVEMENT	
3545	3428	.	.	. SPC005 EQU \$	
3546	3428	D8	.	. RC ;RETURN ON ERROR	
3547	3429	21	D3	3A LXI H,SKPMSG ;SET SKIP LINES MESSAGE	
3548	342C	CD	F0	2D CALL SLTPMS ;SELECT TAPE UNIT MESSAGE	
3549	342F	.	.	. ;*****	
3550	342F	.	.	. ; WAIT FOR SPACING TO BE FINISHED - ENTRY FOR *	
3551	342F	.	.	. ; MONITORING HIGH-SPEED SEARCH *	
3552	342F	.	.	. ;*****	
3553	342F	.	.	. SPCWAT EQU \$;ENTRY FOR MESSAGE UPDATE	
3554	342F	21	29	FF LXI H,CTISTA ;SET COMMAND SOURCE FLAG	
3555	3432	36	FF	. MVI M,-1 ;FOR KEYBOARD SOURCE	
3556	3434	3A	6E	FF LDA DFLGS ;GET DATA TRANSFER FLAGS	
3557	3437	E6	01	. ANI SDACOM ;COMMAND FROM KEYBOARD?	
3558	3439	C8	.	. RZ ;YES - RETURN	
3559	343A	.	.	. SPCWT1 EQU \$;(ENTRY FOR FIND EVD)	
3560	343A	36	00	. MVI M,0 ;NO - SET FOR DATA COMM IN	
3561	343C	CD	A6	3C CALL CARDIS ;DISPLAY SPACE MESSAGE	
3562	343F	.	.	. SPC010 EQU \$	
3563	343F	CD	C1	29 CALL CTMON1 ;MONITOR TAPE DRIVES	
3564	3442	DA	43	00 JC RSTDSP ;ERROR - QUIT	
3565	3445	CA	54	34 JZ SPC020 ;'RUN' TURNED OFF - QUIT	
3566	3448	CD	2E	48 CALL RETSCN ;RET PRESSED?	
3567	344B	D2	3F	34 JNC SPC010 ;NO - CONTINUE WAITING	
3568	344E	CD	E4	2B CALL STOPTP ;YES - USER INTERRUPT	
3569	3451	CD	45	2C CALL FWDSP1 ;ADVANCE 1 REC TO END IN	
3570	3454	.	.	. ; REASONABLE POSITION	
3571	3454	.	.	. SPC020 EQU \$	
3572	3454	CD	43	00 CALL RSTDSP ;RESTORE DISPLAY	
3573	3457	3A	4F	FF LDA IOCERR ;'RUN' OFF, ANY ERROR?	
3574	345A	FE	53	. CPI S	
3575	345C	C8	.	. RZ ;NO - RET	

13255
2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 104
=====
3576     345D     37 . .      STC                                ;YES - C => ERROR
3577     345E     C9 . .      RET
```

=====					PAGE 105	
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	
=====						
3579	345F	.	.	.	;*****	
3580	345F	.	.	.	; TERMINATE SPACE/SEARCH OPERATION *	
3581	345F	.	.	.	;*****	
3582	345F	.	.	.	SPCEN1 EQU \$	
3583	345F	CD	E5	2D	CALL CTUERR ;CHECK FOR I/O ERROR	
3584	3462	.	.	.	SPCEND EQU \$	
3585	3462	3A	29	FF	LDA CTISTA ;GET COMMAND SOURCE FLAG	
3586	3465	B7	.	.	ORA A ;COMMAND FROM KEYBOARD?	
3587	3466	C8	.	.	RZ ;NO - RETURN	
3588	3467	C3	84	36	JMP USREXT ;YES - EXIT THRU USER EXIT	
3589	346A	.	.	.	;*****	
3590	346A	.	.	.	; SPCPFL - SPACE OVER P FILES OR LOCATE FILE P *	
3591	346A	.	.	.	;*****	
3592	346A	.	.	.	SPCPFL EQU \$	
3593	346A	2A	D5	FF	LHLD IOCCNT ;GET PARAMETER VALUE	
3594	346D	3A	DC	FF	LDA IOPSGN ;GET PARAMETER SIGN	
3595	3470	87	.	.	ADD A ;NEGATIVE ADJUSTMENT?	
3596	3471	F2	7F	34	JP SPF100 ;NO - DO NOT DECREMENT	
3597	3474	CD	FC	2A	CALL CHKEOF ;AT END OF FILE?	
3598	3477	C2	7F	34	JNZ SPF100 ;YES - DO NOT DECREMENT	
3599	347A	B5	.	.	ORA L ;NO - PARAMETER = 0?	
3600	347B	CA	7F	34	JZ SPF100 ;YES - DO NOT DECREMENT	
3601	347E	2B	.	.	DCX H ;NO - DECREMENT PARAMETER	
3602	347F	.	.	.	SPF100 EQU \$	
3603	347F	26	00	.	MVI H,0 ;CLEAR HIGH BYTE	
3604	3481	22	DE	FF	SHLD IODATA ;SAVE MODIFIED PARAM. VALUE	
3605	3484	21	5E	FF	LXI H,FILNUM	
3606	3487	46	.	.	MOV B,M ;GET CURRENT FILE COUNT	
3607	3488	0E	FF	.	MVI C,255 ;GET MAXIMUM FILE COUNT	
3608	348A	11	2D	FF	LXI D,CTICNT+1 ;TARGET ADDRESS	
3609	348D	CD	52	00	CALL CHKLIM ;GET ABSOLUTE FILE COUNT	


```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 107
=====
3661      34D6      CD  A5  32          CALL CLRCT0      ;CLEAR EOF INHIBIT
3662      34D9      78  .   .          MOV  A,B         ;GET COMMAND
3663      34DA      .   .   .          SRC110 EQU $
3664      34DA      21  E6  34          LXI  H,TISRC0   ;SET INTERRUPT TO COUNT GAP
3665      34DD      CD  78  2A          CALL OUTCMD     ;ISSUE COMMAND
3666      34E0      .   .   .          ;
3667      34E0      .   .   .          ; SET UP SEARCHING MESSAGE
3668      34E0      .   .   .          ;
3669      34E0      21  37  3B          LXI  H,LOCMSG   ;SET FIRST PART OF MESSAGE
3670      34E3      C3  F0  2D          JMP  SLTPMS     ;SELECT MESSAGE FOR TAPE UNI
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 108
3672	34E6	.	.	*****	
3673	34E6	.	.	; COUNT GAP LENGTH - INTERRUPT ROUTINE *	
3674	34E6	.	.	*****	
3675	34E6	.	.	TISRC0 EQU \$	
3676	34E6	E6	40	ANI LP ;BACKED PAST LP?	
3677	34E8	CA	40 35	JZ SRC500 ;YES - RECOVER	
3678	34EB	7E	.	MOV A,M ;GET CTSTAT	
3679	34EC	E6	20	ANI GAP ;FOUND DATA YET?	
3680	34EE	C2	85 35	JNZ SRC600 ;NO - CHECK FOR EVD	
3681	34F1	.	.	;	
3682	34F1	.	.	; DATA FOUND - PROCESS GAP	
3683	34F1	.	.	;	
3684	34F1	21	75 35	LXI H,TISRC1 ;SET UP INTERRUPT TO WAIT	
3685	34F4	22	E1 FF	SHLD CTIVEC ;FOR GAP	
3686	34F7	CD	15 2C	CALL GAPTST ;UPDATE FILE STATUS & NUMBER	
3687	34FA	.	.	; BASED ON GAP LENGTH	
3688	34FA	21	61 FF	LXI H,RELTAK ;CLEAR GAP COUNTER	
3689	34FD	36	00	MVI M,0	
3690	34FF	D8	.	RC ;NO FILE NUMBER CHANGE - RET	
3691	3500	CD	7A 2D	CALL FILCMP ;FILE CHANGE: REACHED TARGET	
3692	3503	C0	.	RNZ ;NO - RETURN	
3693	3504	CD	E7 2A	CALL CHKFWD ;YES - WHICH DIRECTION?	
3694	3507	C4	9A 35	CNZ SRC700 ;FWD - STOP TAPE AND GO BACK	
3695	350A	CD	E4 2B	CALL STOPTP ;REV - STOP THE TAPE	
3696	350D	97	.	SUB A ;SET TRANSFER LIMIT AT EOF	
3697	350E	32	47 FF	STA XFRLIM	
3698	3511	C5	.	PUSH B ;SAVE REGISTERS REQ'D FOR	
3699	3512	D5	.	PUSH D ;CTU READ	
3700	3513	.	.	SRC310 EQU \$	
3701	3513	CD	DC 2A	CALL GTCTBT ;GET BIT FOR THIS UNIT	
3702	3516	47	.	MOV B,A ;UNIT BIT IN B-REG	
3703	3517	2A	2D FF	LHLD CTICNT+1 ;SAVE DESIRED FILE AND	
3704	351A	E5	.	PUSH H ;RETRY COUNTER	
3705	351B	CD	12 2E	CALL CT2BUF ;READ A RECORD	
3706	351E	E1	.	POP H	
3707	351F	22	2D FF	SHLD CTICNT+1	
3708	3522	DA	33 35	JC SRC320 ;RETURN ON ERROR	
3709	3525	97	.	SUB A ;MARK BUFFER FREE	
3710	3526	12	.	STAX D	
3711	3527	1B	.	DCX D ;D,E -> TYPE	
3712	3528	1A	.	LDAX D	
3713	3529	B7	.	ORA A ;IS THIS A FILE MARK?	
3714	352A	C2	13 35	JNZ SRC310 ;NO - GET NEXT RECORD	
3715	352D	CD	7A 2D	CALL FILCMP ;YES - CORRECT FILE?	
3716	3530	DA	13 35	JC SRC310 ;PRESENT < NEEDED, CONTINUE	
3717	3533	.	.	; READING	
3718	3533	.	.	SRC320 EQU \$;(ENTRY FOR READ ERROR)	
3719	3533	D1	.	POP D ;(RESTORE REGISTERS)	
3720	3534	C1	.	POP B	
3721	3535	C8	.	RZ ;PRESENT = NEEDED, RET	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 109
3722	3536	D8	.	.	RC ;(READ ERROR EXIT)	
3723	3537	21	2E	FF	LXI H,CTICNT+2	
3724	353A	35	.	.	DCR M ;PRESENT > NEEDED, RETRY?	
3725	353B	21	6A	3C	LXI H,FMSMSG ;("FILE MISSING" MESSAGE)	
3726	353E	CA	5F	34	JZ SPEN1 ;NO - REPORT ERROR AND QUIT	
3727	3541	97	.	.	SUB A ;YES - CLEAR GAP COUNTER	
3728	3542	32	61	FF	STA RELTAK	
3729	3545	3E	05	.	MVI A,RUN+FST ;SEARCH FAST BACKWARD	
3730	3547	21	E6	34	LXI H,TISRC0 ;SET INTERRUPT TO COUNT GAP	
3731	354A	C3	78	2A	JMP OUTCMD	
3732	354D	.	.	.	;	
3733	354D	.	.	.	; HIT LP HOLE - POSSIBLE MISSING FILE	
3734	354D	.	.	.	;	
3735	354D	.	.	.	SRC500 EQU \$;RECOVER FROM HITTING LP	
3736	354D	CD	E0	2B	CALL STPTPO ;STOP THE TAPE	
3737	3550	97	.	.	SUB A ;CLEAR CONTROL BITS	
3738	3551	32	62	FF	STA CNTRL0	
3739	3554	3C	.	.	INR A ;SET AT FILE 1	
3740	3555	32	5E	FF	STA FILNUM	
3741	3558	21	2E	FF	LXI H,CTICNT+2	
3742	3558	35	.	.	DCR M ;RETRY?	
3743	355C	21	6A	3C	LXI H,FMSMSG ;("FILE MISSING" MESSAGE)	
3744	355F	CA	5F	34	JZ SPEN1 ;NO - REPORT ERROR AND EXIT	
3745	3562	CD	AD	2B	CALL CHKLPD ;MOVE TO LOAD POINT	
3746	3565	DA	62	34	JC SPEND ;EXIT ON ERROR	
3747	3568	3E	07	.	MVI A,RUN+FWD+FST ;SEARCH FAST FORWARD	
3748	356A	CD	DA	34	CALL SRC110 ;START SEARCH ROUTINE	
3749	356D	3A	29	FF	LDA CTISTA ;GET COMMAND SOURCE FLAG	
3750	3570	B7	.	.	ORA A ;COMMAND FROM DATA COMM?	
3751	3571	C0	.	.	RNZ ;NO - RETURN	
3752	3572	C3	A6	3C	JMP CARDIS ;YES - DISPLAY SEARCH MESSAG	
3753	3575	.	.	.	*****	
3754	3575	.	.	.	; WAIT FOR GAP - INTERRUPT ROUTINE *	
3755	3575	.	.	.	*****	
3756	3575	.	.	.	TISRC1 EQU \$	
3757	3575	E6	40	.	ANI LP ;BACKED PAST LP?	
3758	3577	CA	4D	35	JZ SRC500 ;YES - RECOVER	
3759	357A	7E	.	.	MOV A,M ;GET CTSTAT	
3760	357B	E6	20	.	ANI GAP ;IN GAP?	
3761	357D	C8	.	.	RZ ;NO - CONTINUE TO WAIT	
3762	357E	21	E6	34	LXI H,TISRC0 ;YES - SET UP INTERRUPT TO	
3763	3581	22	E1	FF	SHLD CTIVEC ;COUNT GAP LENGTH	
3764	3584	C9	.	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 110
3766	3585	.	.	*****	
3767	3585	.	.	; CHECK FOR EVD - SHARED BY SEARCH AND SPACE *	
3768	3585	.	.	*****	
3769	3585	.	.	SRC600 EQU \$	
3770	3585	2E	61	MVI L,RELTAK ;INCREMENT GAP COUNT	
3771	3587	34	.	INR M ;REACHED EVD LIMIT?	
3772	3588	F0	.	RP ;NO - RETURN	
3773	3589	CD	E7 2A	CALL CHKFWD ;YES - GOING FORWARD?	
3774	358C	C8	.	RZ ;NO - CONTINUE	
3775	358D	2E	62	MVI L,CNTRL0*256/256	
3776	358F	36	02	MVI M,EVD ;YES - MARK EVD STATUS	
3777	3591	CD	9A 35	CALL SRC700 ;STOP TAPE & GO BACK	
3778	3594	21	1A 3B	LXI H,EVDMSG ;REPORT END OF DATA	
3779	3597	C3	5F 34	JMP SPCEN1 ;REPORT ERROR AND EXIT	
3780	359A	.	.	*****	
3781	359A	.	.	; STOP TAPE AND GO BACK OVER STOPPING DISTANCE *	
3782	359A	.	.	; USED BY: SEARCH AND SPACE, WHEN HITTING EVD *	
3783	359A	.	.	; SEARCH, AFTER FAST FORWARD TO FILE *	
3784	359A	.	.	*****	
3785	359A	.	.	SRC700 EQU \$	
3786	359A	21	00 00	LXI H,0 ;CLEAR CTIADR TO COUNT TACH	
3787	359D	22	33 FF	SHLD CTIADR ;WHILE STOPPING	
3788	35A0	21	C3 35	LXI H,TISRC2 ;SET UP INTERRUPT TO STOP	
3789	35A3	.	.	; TAPE AND COUNT TACH	
3790	35A3	CD	E7 2B	CALL STPTP1 ;STOP TAPE	
3791	35A6	21	33 FF	LXI H,CTIADR ;TAPE STOPPED - ADD A FEW	
3792	35A9	7E	.	MOV A,M ;TACH EDGES TO BE SURE	
3793	35AA	C6	05	ADI 5	
3794	35AC	2F	.	CMA ;COMPLEMENT COUNT TO COUNT	
3795	35AD	77	.	MOV M,A ;DOWN WHILE BACKING UP	
3796	35AE	23	.	INX H ;GET HIGH BYTE	
3797	35AF	7E	.	MOV A,M	
3798	35B0	CE	00	ACI 0 ;ADD ANY CARRY FROM LOW BYTE	
3799	35B2	2F	.	CMA	
3800	35B3	77	.	MOV M,A	
3801	35B4	3E	01	MVI A,RUN ;RETURN TO INITIAL POSITION	
3802	35B6	21	C3 35	LXI H,TISRC2	
3803	35B9	CD	78 2A	CALL OUTCMD	
3804	35BC	.	.	SRC720 EQU \$;WAIT FOR TAPE TO FINISH	
3805	35BC	CD	C1 29	CALL CTMON1 ;MONITOR TAPES	
3806	35BF	C2	BC 35	RNZ SRC720 ;WAIT IF STILL RUNNING	
3807	35C2	C9	.	RET	
3808	35C3	.	.	*****	
3809	35C3	.	.	; DUAL-PURPOSE INTERRUPT ROUTINE: *	
3810	35C3	.	.	; FORWARD: COUNT TACH EDGES IN CTIADR *	
3811	35C3	.	.	; REVERSE: COUNT TACH EDGES UNTIL CTIADR = 0,*	
3812	35C3	.	.	; THEN STOP TAPE *	
3813	35C3	.	.	*****	
3814	35C3	.	.	TISRC2 EQU \$	
3815	35C3	7E	.	MOV A,M ;TEST CTSTAT	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 111
=====
3816     35C4     B7 . .      ORA  A
3817     35C5     F0 . .      RP                               ;NO - RETURN
3818     35C6     2E 33 .     MVI  L,CTIADR ;YES - COUNT
3819     35C8     34 . .      INR  M
3820     35C9     C0 . .      RNZ                               ;RETURN ON NO OVERFLOW
3821     35CA     23 . .      INX  H                           ;OVERFLOW - INC HIGH BYTE
3822     35CB     34 . .      INR  M                           ;CTIADR = 0? (OCCURS ONLY IN
3823     35CC     . . .      ;                                REVERSE)
3824     35CC     C0 . .      RNZ                               ;NO - RETURN
3825     35CD     C3 E0 2B    JMP  STPTP0 ;YES - CLEAR RELTAK & RET
=====
  
```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 112
=====
3827      35D0      . . .      ;*****
3828      35D0      . . .      ; SREVD - SEARCH FOR EVD MARK *
3829      35D0      . . .      ;*****
3830      35D0      . . .      SREVD EQU $
3831      35D0      21 FF FF    LXI H,-1      ;START SPACING MAX. DISTANCE
3832      35D3      CD 57 2C    CALL FWDSPX
3833      35D6      21 14 3B    LXI H,FEVDMS ;GET "FIND EVD" MESSAGE
3834      35D9      CD F0 2D    CALL SLTPMS   ;SET UP WITH UNIT MSG
3835      35DC      21 29 FF    LXI H,CTISTA ;SET UP FOR SPCWAT
3836      35DF      CD 3A 34    CALL SPCWT1   ;WAIT FOR TAPE TO STOP
3837      35E2      CD 0E 2B    CALL CHKEVO   ;AT END OF VALID DATA?
3838      35E5      37 . .     STC
3839      35E6      C8 . .     RZ            ;NO - RETURN ERROR
3840      35E7      CD 97 3D    CALL IOERCL   ;YES - CLEAR ERROR
3841      35EA      3E 08 .     MVI A,CMDEXC ;AND SET "COMMAND EXECUTED
3842      35EC      C3 98 28    JMP STUNT0
3843      35EF      CD 0E 2B    CALL CHKEVO   ;AT END OF VALID DATA?
3844      35F2      C2 97 3D    JNZ IOERCL    ;YES - CLEAR ERROR AND RET
3845      35F5      37 . .     STC            ;NO - RETURN ERROR
3846      35F6      C9 . .     RET
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
3848	35F7	.	.	;*****
3849	35F7	.	.	; TSTCTU - RELIABILITY TEST OF CTU *
3850	35F7	.	.	; *
3851	35F7	.	.	; EXIT: C => TEST FAILURE *
3852	35F7	.	.	; NC => TEST SUCCESS *
3853	35F7	.	.	;*****
3854	35F7	.	.	TSTCTU EQU \$
3855	35F7	3A	F3 FF	LDA MDFLG2 ;GET CURRENT MODE
3856	35FA	47	.	MOV B,A
3857	35FB	C5	.	PUSH B ;SAVE MODE
3858	35FC	CD	D4 36	CALL STWBSR ;SET WRITE/BACKSPACE/READ
3859	35FF	CD	08 36	CALL TCT005 ;DO TEST
3860	3602	C1	.	POP B ;RESTURE FORMER MODE
3861	3603	78	.	MOV A,B
3862	3604	32	F3 FF	STA MDFLG2
3863	3607	C9	.	RET
3864	3608	.	.	TCT005 EQU \$
3865	3608	CD	FF 3C	CALL GTIOB0 ;GET I/O BUFFER
3866	360B	D8	.	RC ;RETURN ON ERROR
3867	360C	2B	.	DCX H
3868	360D	2B	.	DCX H ;H,L -> LENGTH
3869	360E	36	00 .	MVI M,0 ;0 => 256
3870	3610	23	.	INX H ;H,L => TYPE
3871	3611	36	FF .	MVI M,-1 ;-1 => NORMAL
3872	3613	23	.	INX H ;H,L -> STATUS
3873	3614	CD	DC 2A	CALL GTCTBT ;GET FLAG FOR SELECTED UNIT
3874	3617	77	.	MOV M,A
3875	3618	47	.	MOV B,A ;SAVE FOR BUF2CT
3876	3619	EB	.	XCHG ;D,E -> STATUS
3877	361A	CD	2A 3D	CALL GETPTR ;GET PTR TO 1ST BYTE OF BUF
3878	361D	3E	25 .	MVI A,45Q ;WORST CASE PATTERN
3879	361F	.	.	TCT030 EQU \$
3880	361F	77	.	MOV M,A ;PUT BYTE IN BUFFER
3881	3620	EE	7F .	XRI 177Q ;ALTERNATE CHARACTER
3882	3622	2C	.	INR L ;256 BYTES?
3883	3623	C2	1F 36	JNZ TCT030 ;NO - CONTINUE
3884	3626	CD	F7 2F	CALL BUF2CT ;RECORD BUFFER
3885	3629	D8	.	RC ;RETURN ON ERROR
3886	362A	3A	62 FF	LDA CNTRL0 ;ANYTHING BUT RECORD
3887	362D	E6	2F .	ANI EVD+HRDERR+SFTERR+EOF+WRTERR
3888	362F	21	68 3B	LXI H,FAILMS ;(GET FAIL MESSAGE)
3889	3632	C2	E5 2D	JNZ CTUERR ;IF ERROR, RETURN
3890	3635	CD	3D 2D	CALL EOFC ;RECORD FILE MARK
3891	3638	D8	.	RC ;RETURN ON ERROR
3892	3639	3A	63 FF	LDA UNIT0 ;PAST EW HOLE?
3893	363C	E6	80 .	ANI EW
3894	363E	C8	.	RZ ;NO - QUIT
3895	363F	.	.	TCT050 EQU \$;YES - REWIND TAPE
3896	363F	CD	75 2B	CALL RWDLP ;START REWIND OPERATION
3897	3642	.	.	TCT060 EQU \$;WAIT UNTIL REWIND FINISHED

13255

2648A MICROCODE LISTING 'I0273'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
3898     3642     CD  C1  29          CALL CTMON1      ;TAPE STILL RUNNING?
3899     3645     D8  .   .          RC              ;RETURN ON ERROR
3900     3646     C2  42  36          JNZ  TCT060     ;YES - WAIT
3901     3649     C3  08  36          JMP  TCT005     ;LET GTIOBF & BUF2CT WAIT
=====
```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 115
=====
3903      364C      . . .      ;*****
3904      364C      . . .      ; CONDTN - CONDITION TAPE *
3905      364C      . . .      ;*****
3906      364C      . . .      CONDTN EQU $
3907      364C      CD 97 3D    CALL IOERCL      ;CLEAR ERROR (IOCERR <- S)
3908      364F      3E 04 .    MVI A,4          ;CONTROL CODE FOR CONDITION
3909      3651      CD 40 39    CALL USRNPO      ;GET DEVICE AND GO
3910      3654      C3 84 36    JMP USREXT       ;QUIT
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 116
3912	3657	.	.	*****	
3913	3657	.	.	; KEY INTERPRETATION *****	
3914	3657	.	.	*****	
3915	3657	.	.	;	
3916	3657	.	.	; * * * * *	
3917	3657	.	.	;	
3918	3657	.	.	; GRNKEY - USER PRESSED GREEN KEY - PERFORM	
3919	3657	.	.	; I/O OPERATION	
3920	3657	.	.	;	
3921	3657	.	.	; ENTRY: DON'T CARE	
3922	3657	.	.	;	
3923	3657	.	.	; EXIT : NC => NO ERROR	
3924	3657	.	.	; IOCERR = S	
3925	3657	.	.	; C => ERROR	
3926	3657	.	.	; IOCERR = U => USER INTERRUPT	
3927	3657	.	.	; IOCERR = F => FAILURE	
3928	3657	.	.	;	
3929	3657	.	.	;	
3930	3657	.	.	GRN000 EQU \$;SOUND BELL FOR BAD KEY	
3931	3657	CD	14 48	CALL ZBELL	
3932	365A	.	.	GRNKEY EQU \$	
3933	365A	CD	6F 29	CALL CTMON ;MONITOR THE TAPES	
3934	365D	CD	87 3D	CALL IOCCLR ;CLEAR ALL I/O VARIABLES	
3935	3660	36	00 .	MVI M,0 ;RESET ESCAPE FLAG	
3936	3662	CD	05 48	CALL ZGETKY ;ANY INPUT FROM KEYBOARD?	
3937	3665	C2	5A 36	JNZ GRNKEY ;NO - CONTINUE WAITING	
3938	3668	.	.	*****	
3939	3668	FE	EF .	CPI SFTCR ;YES--IS IT RETURN KEY?	
3940	366A	.	.	*****	
3941	366A	C8	. .	RZ ;YES - QUIT	
3942	366B	B7	. .	ORA A ;IS IT NULL?	
3943	366C	CA	57 36	JZ GRN000 ;YES - REPORT ERROR	
3944	366F	.	.	*****	
3945	366F	.	.	; LOOK UP KEY IN VECTOR TABLE *	
3946	366F	.	.	*****	
3947	366F	21	9D 36	LXI H,GRNTBL-3 ;BASE ADDRESS	
3948	3672	01	03 00	LXI B,3 ;DISTANCE BETWEEN ENTRIES	
3949	3675	.	.	GRN100 EQU \$	
3950	3675	09	. .	DAD B ;POINT TO NEXT ENTRY	
3951	3676	BE	. .	CMP M ;IS KEY >= TABLE ENTRY?	
3952	3677	DA	75 36	JC GRN100 ;NO - TRY NEXT ENTRY	
3953	367A	C2	57 36	JNZ GRN000 ;YES - IF NOT SAME, RING BEL	
3954	367D	23	. .	INX H ;VALID COMMAND - GET VECTOR	
3955	367E	5E	. .	MOV E,M	
3956	367F	23	. .	INX H	
3957	3680	56	. .	MOV D,M	
3958	3681	04	. .	INR B ;ROUTINES WANT B = 1	
3959	3682	EB	. .	XCHG	
3960	3683	CF	. .	RST RSTJMP ;DO INDIRECT CALL	
3961	3684	.	.	*****	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 117
3962	3684	.	.	. ; RETURN FOR USER INITIATED I/O FUNCTIONS *	
3963	3684	.	.	. ;*****	
3964	3684	.	.	. USREXT EQU \$	
3965	3684	DC	17 3D	CC FREBFS ;FREE BUFFERS ON ERROR	
3966	3687	3A	4F FF	LDA IOCERR ;ANY ERROR?	
3967	368A	FE	53 .	CPI S	
3968	368C	C8	. .	RZ ;NO - RETURN	
3969	368D	FE	55 .	CPI U ;USER INTERRUPT?	
3970	368F	37	. .	STC ;(C => ERROR)	
3971	3690	C8	. .	RZ ;YES - QUIT	
3972	3691	.	.	. ;*****	
3973	3691	.	.	. ; FAILURE - DISPLAY MESSAGE UNTIL CR REC'D *	
3974	3691	.	.	. ;*****	
3975	3691	CD	A6 3C	CALL CARDIS ;DISPLAY MESSAGE	
3976	3694	.	.	. UTX100 EQU \$	
3977	3694	CD	C1 29	CALL CTMON1 ;CHECK FOR REMOVED TAPES	
3978	3697	CD	2E 48	CALL RETSCN ;RETURN KEY PRESSED?	
3979	369A	D2	94 36	JNC UTX100 ;NO - CONTINUE WAITING	
3980	369D	C3	43 00	JMP RSTDSP ;YES - RESTORE DISP & RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 118
3982	36A0	.	.	;*****	
3983	36A0	.	.	;	
3984	36A0	.	.	; TABLE FOR GREEN KEY FUNCTIONS	
3985	36A0	.	.	;	
3986	36A0	.	.	;*****	
3987	36A0	.	.	GRNTBL EQU \$	
3988	36A0	FF	.	DB 377Q ;CONTROL F1 (COPY ALL)	
3989	36A1	BF	32	DW USRCMA ;COMPARE TO END-OF-DATA	
3990	36A3	FA	.	DB 372Q ;TEST KEY	
3991	36A4	E4	36	DW CTUTST ;PERFORM COMPLETE TEST	
3992	36A6	F7	.	DB 367Q ;F8	
3993	36A7	7C	38	DW USRFFL ;FIND FILE	
3994	36A9	F6	.	DB 366Q ;F7	
3995	36AA	83	38	DW USRSKP ;SKIP LINES	
3996	36AC	F5	.	DB 365Q ;F6	
3997	36AD	3E	39	DW USREOF ;MARK FILE	
3998	36AF	F4	.	DB 364Q ;F5	
3999	36B0	3A	39	DW USRRWD ;REWIND	
4000	36B2	F3	.	DB 363Q ;F4	
4001	36B3	5B	39	DW USRTED ;TOGGLE EDIT MODE	
4002	36B5	F2	.	DB 362Q ;F3	
4003	36B6	CD	3A	DW USRXFL ;COPY LINE	
4004	36B8	F1	.	DB 361Q ;F2	
4005	36B9	CE	3A	DW USRXFF ;COPY FILE	
4006	36BB	F0	.	DB 360Q ;F1	
4007	36BC	CF	3A	DW USRXFA ;COPY ALL	
4008	36BE	DD	.	DB 335Q ;CONTROL F3 (COPY LINE)	
4009	36BF	BD	32	DW USRCMR ;COMPARE LINE	
4010	36C1	DB	.	DB 333Q ;CONTROL F2 (COPY FILE)	
4011	36C2	BE	32	DW USRCMF ;COMPARE FILE	
4012	36C4	CD	.	DB 315Q ;DELETE LINE	
4013	36C5	DC	36	DW CLWBSR ;CLR WRITE/BACKSP/READ MOD	
4014	36C7	CC	.	DB 314Q ;INSERT LINE	
4015	36C8	D4	36	DW STWBSR ;SET WRITE/BACKSP/READ MOD	
4016	36CA	A0	.	DB 240Q ;CONTROL READ	
4017	36CB	78	32	DW EVDRED ;READ BEYOND END-OF-DATA	
4018	36CD	98	.	DB 230Q ;ENTER KEY	
4019	36CE	A2	00	DW DCTEST ;DATACOM SELF-TEST	
4020	36D0	20	.	DB 40Q ;SPACE BAR	
4021	36D1	DB	3F	DW TPSTAT ;DISPLAY FILE, INCHES	
4022	36D3	00	.	DB 0	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 119
=====
4024     36D4      . . .      STWBSR EQU $ ;SET WRITE/BS/READ MODE
4025     36D4      21 F3 FF    LXI H,MDFLG2
4026     36D7      7E . .     MOV A,M
4027     36D8      F6 20 .    ORI WBSR
4028     36DA      77 . .     MOV M,A
4029     36DB      C9 . .     RET
4030     36DC      . . .      CLWBSR EQU $ ;CLEAR WRITE/BS/READ MODE
4031     36DC      21 F3 FF    LXI H,MDFLG2
4032     36DF      7E . .     MOV A,M
4033     36E0      E6 DF .    ANI -1-WBSR
4034     36E2      77 . .     MOV M,A
4035     36E3      C9 . .     RET
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 120
=====
4037      36E4      . . .      ;
4038      36E4      . . .      ; * * * * *
4039      36E4      . . .      ;
4040      36E4      . . .      ;          CTUTST - DO COMPLETE TEST OF TERMINAL
4041      36E4      . . .      ;          INCLUDING TEST OF BOTH TAPE UNITS.
4042      36E4      . . .      ;
4043      36E4      . . .      ;          TESTS LEFT CTU, THEN TESTS TERMINAL TWICE,
4044      36E4      . . .      ;          THEN RIGHT CTU, THEN TERMINAL ONCE.
4045      36E4      . . .      ;
4046      36E4      . . .      ;
4047      36E4      . . .      CTUTST EQU $
4048      36E4      CD 97 3D    CALL IOERCL      ;CLEAR I/O ERROR FLAG
4049      36E7      3E 07 .    MVI A,7         ;7 = CODE FOR TAPE TEST
4050      36E9      32 D8 FF    STA IOCTYP
4051      36EC      CD E8 33    CALL CTTLLCT    ;CONTROL ROUTINE FOR LEFT TA
4052      36EF      DA FE 36    JC CTT100      ;DISPLAY AND HANG ON ERROR
4053      36F2      CD 7F 00    CALL TEST      ;TEST TERMINAL
4054      36F5      CD 7F 00    CALL TEST      ;TEST TERMINAL
4055      36F8      CD FD 33    CALL CTLRCT    ;CONTROL ROUT. FOR RIGHT TAP
4056      36FB      D2 7F 00    JNC TEST      ;IF OK, RETEST TERMINAL & RE
4057      36FE      . . .      CTT100 EQU $
4058      36FE      2A F1 FF    LHLD MSGPT1    ;NOT OK - SET UP FOR HANGUO
4059      3701      C3 9D 00    JMP HANGUO     ;DISPLAY MSG UNTIL HARD RESE
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
4061	3704	.	.	*****
4062	3704	.	.	; SELKEY - DEVICE SELECTION *
4063	3704	.	.	*****
4064	3704	.	.	SELKEY EQU \$
4065	3704	3A	F4 FF	LDA MOFLG1 ;IN EDIT OR LOGGING MODE?
4066	3707	E6	10 .	ANI EDIT
4067	3709	C2	14 48	JNZ ZBELL ;YES - CAN'T CHANGE DEVICES
4068	370C	3E	20 .	MVI A,SELECT
4069	370E	06	00 .	MVI B,0
4070	3710	CD	0E 48	CALL ZSTMD1 ;TURN ON SELECT MODE
4071	3713	CD	87 3D	CALL IOCCLR ;CLEAR I/O STORAGE
4072	3716	36	00 .	MVI M,0 ;RESET ESCAPE FLAG
4073	3718	.	.	SLK050 EQU \$
4074	3718	CD	6F 29	CALL CTMON ;MONITOR TAPES
4075	3718	CD	05 48	CALL ZGETKY ;ANY KEYBOARD INPUT?
4076	371E	32	9C FF	STA CHARIN
4077	3721	C2	18 37	JNZ SLK050 ;NO - CONTINUE SCANNING
4078	3724	FE	9E .	CPI SLKYCD ;IS IT THE SELECT KEY?
4079	3726	CA	5C 37	JZ SLK410 ;YES - ABORT DEVICE SELECT
4080	3729	FE	F3 .	CPI 3630 ;IS IT F4 (PRINTER INPUT)?
4081	372B	CA	18 37	JZ SLK050 ;YES - IGNORE IT
4082	372E	CD	1F 39	CALL CKDVKY ;DEVICE KEY?
4083	3731	C2	51 37	JNZ SLK200 ;NO - TERMINATE PROCESSING
4084	3734	4F	. .	MOV C,A ;SAVE DEVICE FLAG
4085	3735	3A	9C FF	LDA CHARIN ;RECALL THE CHARACTER
4086	3738	21	D9 FF	LXI H,IOCINP ;(SET ADDRESS FOR INPUT)
4087	373B	FE	CC .	CPI 3140 ;ALTERNATE I/O INPUT?
4088	373D	CA	4B 37	JZ SLK150
4089	3740	FE	F0 .	CPI 3600 ;OTHER INPUT SPEC?
4090	3742	DA	4A 37	JC SLK130 ;NO - ALTERNATE I/O OUTPUT
4091	3745	E6	04 .	ANI 40 ;OTHER INPUT SPEC?
4092	3747	CA	4B 37	JZ SLK150 ;YES -
4093	374A	.	.	SLK130 EQU \$
4094	374A	23	. .	INX H ;NO - POINT TO OUTPUT
4095	374B	.	.	SLK150 EQU \$
4096	374B	79	. .	MOV A,C ;RECALL DEVICE FLAG
4097	374C	B6	. .	ORA M ;ACCUMULATE DEVICE FLAGS
4098	374D	77	. .	MOV M,A ;SAVE FLAGS
4099	374E	C3	18 37	JMP SLK050 ;GET NEXT CHAR.
4100	3751	.	.	;
4101	3751	.	.	; NON-DEVICE KEY HIT - TERMINATE SELECTION MODE
4102	3751	.	.	;
4103	3751	.	.	SLK200 EQU \$
4104	3751	CD	64 37	CALL SETDEV ;VALID ASSIGNMENT?
4105	3754	DC	E5 2D	CC CTUERR ;NO - FLAG ERROR
4106	3757	3E	03 .	MVI A,RPTKEY
4107	3759	D4	08 48	CNC ZKBCTL ;YES - PUT CHAR BACK INTO IN
4108	375C	.	.	SLK410 EQU \$;ABORT ENTRY
4109	375C	3E	20 .	MVI A,SELECT ;TURN OFF LIGHT
4110	375E	CD	11 48	CALL ZCLMD1

13255

2648A MICROCODE LISTING 'I0273'

13255/90010

REV 04/17/78

=====

ITEM LOC OBJECT CODE SOURCE STATEMENTS

PAGE 122

=====

4111 3761 C3 84 36 JMP USREXT ;RETURN TO SYSTEM

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
4113	3764	.	.	;
4114	3764	.	.	; * * * * *
4115	3764	.	.	;
4116	3764	.	.	; SETDEV - SET DEVICE ASSIGNMENT
4117	3764	.	.	;
4118	3764	.	.	; ENTRY: IOCINP, IOCCOUT CONTAIN NEW
4119	3764	.	.	; DEVICE ASSIGNMENTS
4120	3764	.	.	;
4121	3764	.	.	; EXIT : C => ERROR
4122	3764	.	.	; H,L -> APPROPRIATE ERROR MSG
4123	3764	.	.	; NC => NO ERROR
4124	3764	.	.	; INPDEV, OUTDEV UPDATED
4125	3764	.	.	; A DESTROYED
4126	3764	.	.	;
4127	3764	.	.	SETDEV EQU \$
4128	3764	21	D9 FF	LXI H,IOCINP ;CHECK INPUT ASSIGNMENT
4129	3767	AF	.	XRA A
4130	3768	96	.	SUB M ;ANY ASSIGNMENT?
4131	3769	CA	7A 37	JZ STD010 ;NO - CHECK OUTPUT ASSIGNMENT
4132	376C	A6	.	ANA M ;YES - MASK FOR RIGHTMOST BIT
4133	376D	BE	.	CMP M ;ONLY ONE ASSIGNMENT?
4134	376E	21	DF 3B	LXI H,TMFMSG ;(TUO MANY "FROM" DEVS)
4135	3771	37	.	STC
4136	3772	C0	.	RNZ ;NO - RETURN FAIL
4137	3773	FE	08 .	CPI PRINTR ;IS IT THE PRINTER?
4138	3775	37	.	STC
4139	3776	C8	.	RZ ;YES - RETURN ERROR
4140	3777	32	4E FF	STA INPDEV ;AND STORE IT
4141	377A	.	.	;
4142	377A	.	.	; CHECK OUTPUT ASSIGNMENT
4143	377A	.	.	;
4144	377A	.	.	STD010 EQU \$
4145	377A	3A	DA FF	LDA IOCCOUT ;NEW OUTPUT ASSIGNMENT?
4146	377D	B7	.	ORA A
4147	377E	C8	.	RZ ;NO - RETURN SUCCESSFUL
4148	377F	32	4D FF	STA OUTDEV ;YES - STORE IT
4149	3782	AF	.	XRA A ;SET Z TO TRUE
4150	3783	C9	.	RET ;RETURN

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 124
=====
4152      3784      . . .      ;*****
4153      3784      . . .      ; CONTROL READ PRESSED *
4154      3784      . . .      ;*****
4155      3784      . . .      CTLRED EQU $
4156      3784      21 F3 FF      LXI H,MDFLG2 ;GET HARD MODE FLAGS
4157      3787      7E . .      MOV A,M
4158      3788      E6 08 .      ANI REMOTE ;REMOTE MODE ENABLED?
4159      378A      CA 9F 37      JZ REDKEY ;NO - DO NORMAL READ
4160      378D      CD 97 3D      CALL IOERCL ;YES - CLEAR I/O ERROR FLAG
4161      3790      7E . .      MOV A,M ;RECALL HARD MODE FLAGS
4162      3791      2F . .      CMA
4163      3792      E6 02 .      ANI BLKMDE ;IN BLOCK MODE?
4164      3794      CC B1 00      CZ GTMODE ;YES - CHECK LINE/PAGE
4165      3797      3E 01 .      MVI A,RDWOWT ;IF NOT BLOCK, LINE, DO READ
4166      3799      C4 24 2B      CNZ STIOFS ;WITHOUT WAIT
4167      379C      C3 AC 37      JMP RED010 ;GO TO READ KEY ROUTINE
=====
```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
4169      379F      . . .      ;*****
4170      379F      . . .      ; REDKEY - READ KEY PRESSED *
4171      379F      . . .      ;*****
4172      379F      . . .      REDKEY EQU $
4173      379F      CD 97 3D      CALL I0ERCL      ;CLEAR I/O ERROR FLAG
4174      37A2      3A F3 FF      LDA MDFLG2      ;LOCAL MODE? (REMOTE UP OR
4175      37A5      E6 0A .      ANI REMOTE+BLKMDE ;BLOCK DOWN?)
4176      37A7      EE 08 .      XRI REMOTE
4177      37A9      C2 D4 37      JNZ RED100      ;YES - DO LOCAL FILE COPY
4178      37AC      . . .      ;*****
4179      37AC      . . .      ; REMOTE READ (TO DATACOM) *
4180      37AC      . . .      ;*****
4181      37AC      . . .      RED010 EQU $
4182      37AC      3A FC FF      LDA KBDCSW      ;FULL DUPLEX?
4183      37AF      E6 80 .      ANI FULDUP
4184      37B1      CA B6 37      JZ RED020      ;NO -
4185      37B4      3E 01 .      MVI A,EXTB2D   ;YES - INHIBIT ECHO TO DISP
4186      37B6      . . .      RED020 EQU $
4187      37B6      32 64 FF      STA IOFLG2
4188      37B9      97 . .      SUB A
4189      37BA      32 D8 FF      STA IOCTYP      ;ASCII XFR, NO BYTE COUNT
4190      37BD      32 47 FF      STA XFRLIM      ;TRANSFER TO END OF FILE
4191      37C0      CD C1 2D      CALL BSYCK0     ;INPUT TAPE => WAIT TILL FRE
4192      37C3      D8 . .      RC              ;RETURN ON USER INTERRUPT
4193      37C4      3E 02 .      MVI A,USREAD   ;SET USER READ FLAG
4194      37C6      CD 24 2B      CALL STIOFS
4195      37C9      E6 01 .      ANI RDWOWT     ;READ W/O WAIT?
4196      37CB      C2 BA 41      JNZ IORDGO     ;YES - SEND IMMEDIATELY
4197      37CE      01 01 00      LXI B,SDVREC
4198      37D1      C3 5B 00      JMP SBLXFA
4199      37D4      . . .      ;*****
4200      37D4      . . .      ; LOCAL READ (TO DISPLAY) *
4201      37D4      . . .      ;*****
4202      37D4      . . .      RED100 EQU $
4203      37D4      3A F4 FF      LDA MDFLG1     ;CHECK FOR EDIT MODE
4204      37D7      E6 10 .      ANI EDIT
4205      37D9      CA E3 37      JZ RED120
4206      37DC      CD 64 00      CALL CURPHD    ;EDIT - HOME DOWN
4207      37DF      CD 67 00      CALL FRECNT    ;ENOUGH BLOCKS?
4208      37E2      C0 . .      RNZ           ;NO - RETURN
4209      37E3      . . .      RED120 EQU $
4210      37E3      2A 4D FF      LHLD OUTDEV    ;H <- INPUT DEVICE
4211      37E6      2E 04 .      MVI L,DISPLY   ;SET OUTPUT DEV = DISPLAY
4212      37E8      97 . .      SUB A          ;SET FOR XFR TO END OF FILE
4213      37E9      CD CE 3E      CALL XFR001    ;DO THE READ
4214      37EC      C3 84 36      JMP USREXT     ;REPORT ANY ERRORS AND RET
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS.	PAGE 126
4216	37EF	.	.	*****	
4217	37EF	.	.	; RECKEY - RECORD KEY PRESSED *	
4218	37EF	.	.	*****	
4219	37EF	.	.	RECKEY EQU \$	
4220	37EF	CD	97 3D	CALL IOERCL ;CLEAR I/O ERROR FLAG	
4221	37F2	21	F4 FF	LXI H,MDFLG1 ;IN EDIT MODE?	
4222	37F5	7E	.	MOV A,M	
4223	37F6	E6	10 .	ANI EDIT	
4224	37F8	.	.	*****	
4225	37F8	.	.	; ROM BREAK 2	
4226	37F8	C3	02 38	JMP ZBRK2C	
4227	37FB	.	.	ORG ZBRK1+40000	
4228	3800	.	.	ZBRK2 EQU \$	
4229	3800	54	.	DB VERSN ;ROM PRESENT/VERSION FLAG	
4230	3801	38	.	DB ZBRK2/256	
4231	3802	.	.	ZBRK2C EQU \$	
4232	3802	.	.	*****	
4233	3802	C2	56 3A	JNZ USREDA ;YES - TERMINATE EDIT MODE	
4234	3805	7E	.	MOV A,M ;NO - IN RECORD MODE?	
4235	3806	E6	40 .	ANI RECORD	
4236	3808	C2	60 38	JNZ RCK700 ;YES - END RECORD MODE	
4237	380B	3E	10 .	MVI A,RECINI ;SET "RECORD INIT" FLAG	
4238	380D	CD	24 2B	CALL STIOFS	
4239	3810	3A	F3 FF	LDA MDPLG2 ;REMOTE?	
4240	3813	E6	08 .	ANI REMOTE	
4241	3815	3E	40 .	MVI A,RECORD ;(SET UP TO BLINK RECORD	
4242	3817	06	FF .	MVI B,-1 ;LED)	
4243	3819	C2	0E 48	JNZ ZSTMD1 ;YES - GO INTO RECORD MODE	
4244	381C	.	.	*****	
4245	381C	.	.	; LOCAL RECORD *	
4246	381C	.	.	*****	
4247	381C	CD	5E 00	CALL STRTBL ;FIND START OF BLOCK	
4248	381F	2A	4D FF	LHLD OUTDEV ;L <- OUTPUT DEVICE(S)	
4249	3822	26	04 .	MVI H,DISPLY ;SET INPUT = DISPLAY	
4250	3824	97	.	SUB A ;SET FOR XFR TO END OF DATA	
4251	3825	3C	.	INR A	
4252	3826	CD	CE 3E	CALL XFR001 ;DO THE RECORD	
4253	3829	.	.	TO INITDG	
4254	3829	DA	84 36	JC USREXT ;QUIT ON ERROR	
4255	382C	CD	CC 47	CALL CHKfmt ;IN FORMAT MODE?	
4256	382F	CA	84 36	JZ USREXT ;NO - RETURN TO SYSTEM	
4257	3832	CD	61 00	CALL CURPH ;YES - HOME THE CURSOR AND	
4258	3835	CD	70 00	CALL CLEARS ;CLEAR UNPROTECTED FIELDS	
4259	3838	C3	84 36	JMP USREXT ;EXIT TO SYSTEM	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 127
4261	383B	.	.	*****	
4262	383B	.	.	; REMOTE RECORD *	
4263	383B	.	.	*****	
4264	383B	.	.	;	
4265	383B	.	.	; RECORDING IS TRIGGERED BY CHAR FROM DATACOM	
4266	383B	.	.	; OTHER THAN CR OR LF. WHEN RECEIVED, SYSTEM	
4267	383B	.	.	; CALLS RCRDGO.	
4268	383B	.	.	;	
4269	383B	.	.	RCRDGO EQU \$	
4270	383B	32	88	FF STA CHAR ;SAVE THE FIRST CHAR	
4271	383E	3E	40	. MVI A,RECORD ;TURN OFF LED BLINK	
4272	3840	06	00	. MVI B,0	
4273	3842	CD	0E	48 CALL ZSTMD1	
4274	3845	.	.	RCK650 EQU \$	
4275	3845	CD	FF	3C CALL GTIOB0 ;GET BUFFER	
4276	3848	DA	60	38 JC RCK700 ;TERMINATE MODE ON ERROR	
4277	384B	36	20	. MVI M,DATCOM ;MARK BUF FOR DATACOM INPUT	
4278	384D	2B	.	. DCX H ;H,L -> TYPE	
4279	384E	36	FF	. MVI M,-1 ;ALL RECORDS ARE NORML (DATA	
4280	3850	2B	.	. DCX H ;H,L -> LENGTH	
4281	3851	36	00	. MVI M,0 ;INIT TO 0	
4282	3853	EB	.	. XCHG ;D,E -> LENGTH	
4283	3854	CD	E8	40 CALL DC2BUF ;GET RECORD FROM DATACOM	
4284	3857	.	.	. *****	
4285	3857	.	.	. ; PATCH TO FLUSH BUFFER ON END OF RECORD MODE	
4286	3857	DA	68	38 JC RCK750 ;QUIT RECORD MODE ON ERROR	
4287	385A	CD	76	38 CALL RCK800 ;OUTPUT THE RECORD	
4288	385D	.	.	. *****	
4289	385D	D2	45	38 JNC RCK650 ;NO ERROR - GET NEXT RECORD	
4290	3860	.	.	. RCK700 EQU \$;QUIT RECORD MODE	
4291	3860	CD	17	3D CALL FREBFS ;FREE BUFFERS	
4292	3863	3E	40	. MVI A,RECORD ;TURN OFF LED	
4293	3865	CD	11	48 CALL ZCLMD1	
4294	3868	C3	84	36 JMP USREXT ;SET TRIG, REPORT ERRORS (IF	
4295	386B	.	.	. ; ANY), QUIT	
4296	386B	.	.	. *****	
4297	386B	.	.	. RCK750 EQU \$	
4298	386B	7E	.	. MOV A,M ;FETCH LENGTH	
4299	386C	B7	.	. ORA A ;ANYTHING IN BUFFER?	
4300	386D	CA	60	38 JZ RCK700 ;NO, QUIT	
4301	3870	CD	76	38 CALL RCK800 ;YES-OUTPUT THE RECORD	
4302	3873	C3	60	38 JMP RCK700 ;QUIT	
4303	3876	.	.	. ;	
4304	3876	.	.	. RCK800 EQU \$	
4305	3876	EB	.	. XCHG ;DC2BUF WANTS HL = LENGTH	
4306	3877	13	.	. INX D ;WANT DE = STATUS FOR	
4307	3878	13	.	. INX D ;PUTIO	
4308	3879	C3	7A	41 JMP PUTIO ;OUTPUT THE RECORD	
4309	387C	.	.	. *****	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 128
4311	387C	.	.	. ;*****	
4312	387C	.	.	. ; USRFFL - LOCATE FILE *	
4313	387C	.	.	. ;*****	
4314	387C	.	.	. USRFFL EQU \$	
4315	387C	21	37	3B LXI H,LOCMSG ;DISPLAY LOCATE FILE MESSAGE	
4316	387F	04	.	. INR B ;SET FUNCTION CODE	
4317	3880	C3	86	38 JMP USS010 ;ACCUMULATE PARAMETER	
4318	3883	.	.	. ;*****	
4319	3883	.	.	. ; USRSKP - SPACE OVER RECORDS *	
4320	3883	.	.	. ;*****	
4321	3883	.	.	. USRSKP EQU \$	
4322	3883	21	D3	3A LXI H,SKPMSG ;DISPLAY SPACE MESSAGE	
4323	3886	.	.	. USS010 EQU \$	
4324	3886	3E	0A	. MVI A,DECRDX ;SET RADIX TO DECIMAL	
4325	3888	32	D4	FF STA RADIX	
4326	388B	22	F1	FF SHLD MSGPT1 ;SET PARAMETER MESSAGE	
4327	388E	21	89	3C LXI H,M1MSG ;SET DEFAULT PARAMETER = -1	
4328	3891	22	EF	FF SHLD MSGPT2	
4329	3894	21	D8	FF LXI H,IOCTYP ;STORE FUNCTION TYPE CODE	
4330	3897	70	.	. MOV M,B	
4331	3898	.	.	. USS020 EQU \$	
4332	3898	CD	A6	3C CALL CARDIS ;DISPLAY MESSAGE	
4333	389B	.	.	. USS030 EQU \$	
4334	389B	CD	6F	29 CALL CTMON ;MONITOR TAPES	
4335	389E	CD	05	48 CALL ZGETKY ;ANY KEYBOARD INPUT?	
4336	38A1	C2	9B	38 JNZ USS030 ;NO - CONTINUE SCANNING	
4337	38A4	.	.	. ;*****	
4338	38A4	FE	EF	. CPI SFTCR ;RETURN KEY HIT?	
4339	38A6	.	.	. ;*****	
4340	38A6	CA	43	00 JZ RSTDSP ;YES - ABORT	
4341	38A9	CD	D3	38 CALL USS300 ;DIGIT OR SIGN?	
4342	38AC	D2	98	38 JNC USS020 ;YES - DISPLAY MESSAGE AGAIN	
4343	38AF	CD	1F	39 CALL CKDVKY ;IS IT A DEVICE KEY?	
4344	38B2	C4	14	48 CNZ ZBELL ;NO - SOUND BELL	
4345	38B5	C2	98	38 JNZ USS030 ;AND TRY ANOTHER KEY	
4346	38B8	.	.	. IOSCTL EQU \$	
4347	38B8	32	DB	FF STA IOCDEV ;SAVE DEVICE CODE	
4348	38BB	2A	DE	FF LHLD IODATA ;GET ACCUMULATED PARAMETER	
4349	38BE	3A	DD	FF LDA IOCSGN ;ANY VALUE SPECIFIED?	
4350	38C1	B7	.	. ORA A	
4351	38C2	C2	C7	38 JNZ USS050 ;YES - STORE PARAMETER	
4352	38C5	2F	.	. CMA ;NO - SET DEFAULT OF -1	
4353	38C6	23	.	. INX H ;MAGNITUDE = 1	
4354	38C7	.	.	. ;	
4355	38C7	.	.	. USS050 EQU \$	
4356	38C7	32	DC	FF STA IOPSGN ;STORE SIGN	
4357	38CA	22	D5	FF SHLD IOCCNT ;STORE MAGNITUDE	
4358	38CD	CD	43	00 CALL RSTDSP ;RESTORE NORMAL DISPLAY	
4359	38D0	C3	BE	3F JMP CTR025 ;AND PERFORM FUNCTION	

=====				PAGE 129	
ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS		
=====					
4361	38D3	. . .	;		
4362	38D3	. . .	;	PROCESS DIGIT OR SIGN KEYS	
4363	38D3	. . .	;		
4364	38D3	. . .	USS300 EQU \$		
4365	38D3	FE 2B .	CPI PLUS	;PLUS SIGN?	
4366	38D5	CC 49 00	CZ DCPLUS	;YES - SET SIGN FLAG	
4367	38D8	CA F0 38	JZ USS305	;AND UPDATE MESSAGE	
4368	38D8	FE 2D .	CPI MINUS	;MINUS?	
4369	38DD	CC 4C 00	CZ DCMNUS	;YES - SET SIGN FLAG	
4370	38E0	CA F0 38	JZ USS305	;AND UPDATE MESSAGE	
4371	38E3	FE 30 .	CPI ZERO	;DIGIT?	
4372	38E5	D8 . .	RC	;NO - RETURN	
4373	38E6	FE 3A .	CPI ZERO+10	;DECIMAL DIGIT?	
4374	38E8	32 88 FF	STA CHAR	;(SAVE CHARACTER IN CASE)	
4375	38E8	3F . .	CMC	;(INVERT SENSE OF COMPARE)	
4376	38EC	D8 . .	RC	;NO - RETURN	
4377	38ED	CD 46 00	CALL DCNUM	;YES - ACCUMULATE THE DIGIT	
4378	38F0	. . .	USS305 EQU \$;SET UP MESSAGE	
4379	38F0	3A DD FF	LDA IOCSGN	;GET SIGN VALUE	
4380	38F3	87 . .	ADD A	;ANY SIGN SPECIFIED?	
4381	38F4	21 61 3C	LXI H,BLKMSG	;(SET FOR BLANK MESSAGE)	
4382	38F7	CA 03 39	JZ USS310	;NO - ADD NO SIGN TO MESSAGE	
4383	38FA	21 8D 3C	LXI H,PLSMSG	;(SET FOR PLUS MESSAGE)	
4384	38FD	F2 03 39	JP USS310	;PLUS - DISPLAY PLUS SIGN	
4385	3900	21 90 3C	LXI H,MNSMSG	;MINUS - DISPLAY MINUS SIGN	
4386	3903	. . .	USS310 EQU \$		
4387	3903	22 EF FF	SHLD MSGPT2	;SET SIGN VALUE	
4388	3906	2A DE FF	LHLD IODATA	;GET ACCUMULATED VALUE	
4389	3909	EB . .	XCHG	;PUT INTO D,E	
4390	390A	21 3D FF	LXI H,B2DBUF	;GET OUTPUT BUFFER ADDRESS	
4391	390D	22 ED FF	SHLD MSGPT3	;SET MESSAGE POINTER	
4392	3910	3A D8 FF	LDA IOCTYP	;GET COMMAND TYPE	
4393	3913	3D . .	DCR A	;SEARCH COMMAND?	
4394	3914	7B . .	MOV A,E	;(PUT LSB INTO A-REG)	
4395	3915	C4 AB 00	CNZ BN2DE0	;YES - DISPLAY ONE BYTE ONLY	
4396	3918	CC A8 00	CZ BN2DEC	;NO - DISPLAY TWO BYTE VALUE	
4397	391B	36 CE .	MVI M,EOP	;ADD TERMINATOR CHARACTER	
4398	391D	B7 . .	ORA A	;RETURN "NC" TO INDICATE	
4399	391E	C9 . .	RET	;VALID DIGIT OR SIGN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 130
4401	391F	.	.	;	
4402	391F	.	.	;	
4403	391F	.	.	;	*****
4404	391F	.	.	;	
4405	391F	.	.	;	CKDVKY - CHECK FOR DEVICE KEY
4406	391F	.	.	;	
4407	391F	.	.	;	ENTRY: A = KEY VALUE
4408	391F	.	.	;	
4409	391F	.	.	;	EXIT : Z = T, DEVICE KEY
4410	391F	.	.	;	A = DEVICE BIT
4411	391F	.	.	;	Z = F, NON-DEVICE KEY
4412	391F	.	.	;	A DESTROYED
4413	391F	.	.	;	H,L DESTROYED
4414	391F	.	.	;	
4415	391F	2E	05	CKDVKY EQU \$	
4416	3921	FE	CC	MVI L,5	;FIRST TEST FOR ALTERNATE I/
4417	3923	CA	35 39	CPI 314Q	;INSERT LINE?
4418	3926	FE	D1	JZ CKD100	;YES -
4419	3928	CA	35 39	CPI 321Q	;INSERT CHAR ON?
4420	3928	FE	D2	JZ CKD100	;YES -
4421	392D	CA	35 39	CPI 322Q	;INSERT CHAR OFF?
4422	3930	E6	FB	JZ CKD100	;YES -
4423	3932	D6	EF	ANI 373Q	;F1:=F1+F5, ETC.
4424	3934	6F	.	SUI 357Q	;CHAR.-357B
4425	3935	.	.	MOV L,A	
4426	3935	26	00	CKD100 EQU \$	
4427	3937	C3	B3 3E	MVI H,0	
				JMP DFNDV0	;CHECK VALID DEVICE

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 131
=====
4429     393A      . . .      ;*****
4430     393A      . . .      ; USRRWD - REWIND TAPE TO BOT *
4431     393A      . . .      ;*****
4432     393A      . . .      USRRWD EQU $
4433     393A      AF . .      XRA A          ;SET CONTROL CODE
4434     393B      C3 40 39    JMP USRNPO     ;GET UNIT SELECT
4435     393E      . . .      ;*****
4436     393E      . . .      ; USREOF - RECORD FILE MARK *
4437     393E      . . .      ;*****
4438     393E      . . .      USREOF EQU $
4439     393E      3E 05 .      MVI A,5       ;SET CONTROL CODE
4440     3940      . . .      USRNPO EQU $
4441     3940      32 D8 FF    STA IOCTYP    ;STORE CONTROL CODE
4442     3943      . . .      USRNPM EQU $
4443     3943      CD 6F 29    CALL CTMON    ;MONITOR TAPES
4444     3946      CD 05 48    CALL ZGETKY   ;ANY KEYBOARD INPUT?
4445     3949      C2 43 39    JNZ USRNPM    ;NO - CONTINUE SCANNING
4446     394C      . . .      ;*****
4447     394C      FE EF .      CPI SFTCR     ;RETURN KEY?
4448     394E      . . .      ;*****
4449     394E      C8 . .      RZ            ;YES - ABORT
4450     394F      CD 1F 39    CALL CKOVKY   ;IS IT A DEVICE KEY?
4451     3952      CA B8 38    JZ IOSCTL     ;YES - PERFORM FUNCTION
4452     3955      CD 14 48    CALL ZBELL    ;NO - SOUND THE BELL
4453     3958      C3 43 39    JMP USRNPM    ;GET ANOTHER KEY
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 132
=====
4455     395B      . . .      ;*****
4456     395B      . . .      ; USRTED - TOGGLE EDIT MODE *
4457     395B      . . .      ;*****
4458     395B      . . .      USRTED EQU $
4459     395B      21 F4 FF    LXI H,MDFLG1 ;EDIT MODE ON?
4460     395E      7E . .     MOV A,M
4461     395F      E6 10 .     ANI EDIT
4462     3961      CA 7D 39    JZ UTE030 ;NO - TURN ON EDIT
4463     3964      . . .      ;*****
4464     3964      . . .      ; TURN OFF EDIT MODE *
4465     3964      . . .      ;*****
4466     3964      . . .      EDRST EQU $ ;ENTRY FOR CTU RESET
4467     3964      3E 10 .     MVI A,EDIT ;TURN OFF EDIT LIGHT
4468     3966      CD 11 48    CALL ZCLMD1
4469     3969      21 24 FF    LXI H,SWPCTU ;DATA LOGGING IN SWAP CTU'S
4470     396C      7E . .     MOV A,M ;MODE?
4471     396D      B7 . .     ORA A
4472     396E      C8 . .     RZ ;NO - RETURN
4473     396F      36 00 .     MVI M,0 ;YES - CLEAR SWPCTU
4474     3971      3E F7 .     MVI A,-1-RECRWD ;CLEAR PENDING REWIND
4475     3973      CD 2A 2B    CALL CLIOFS
4476     3976      2E 4D .     MVI L,OUTDEV ;SET UP BOTH CTU'S AS
4477     3978      7E . .     MOV A,M ;OUTPUT DEVICES
4478     3979      F6 03 .     ORI LFTCTU+RGCTU
4479     397B      77 . .     MOV M,A
4480     397C      C9 . .     RET
4481     397D      . . .      ;*****
4482     397D      . . .      ; TURN ON EDIT MODE *
4483     397D      . . .      ;*****
4484     397D      . . .      UTE030 EQU $
4485     397D      7E . .     MOV A,M ;FORMAT MODE?
4486     397E      E6 08 .     ANI FORMAT
4487     3980      C0 . .     RNZ ;YES - IGNORE EDIT REQUEST
4488     3981      2E 4E .     MVI L,INPDEV ;CHECK FOR INPUT/OUTPUT DEV
4489     3983      7E . .     MOV A,M ;DUPLICATION
4490     3984      F6 04 .     ORI DISPLY ;(ADD DISPLAY AS INPUT DEV
4491     3986      2B . .     DCX H ;(GET OUTPUT DEVICES)
4492     3987      A6 . .     ANA M ;ANY DUPLICATION?
4493     3988      C2 AD 41    JNZ IOFAIO ;YES - REPORT FROM = TO
4494     3988      CD 9A 00    CALL MLKOFF ;TURN OFF MEMORY LOCK
4495     398E      06 00 .     MVI B,0 ;SET B FOR NO BLINK (LOCAL)
4496     3990      3A F3 FF    LDA MDFLG2 ;LOCAL MODE?
4497     3993      E6 08 .     ANI REMOTE
4498     3995      CA AB 39    JZ UTE070 ;YES - TURN ON LIGHT & RET
4499     3998      . . .      ;*****
4500     3998      . . .      ; SET UP REMOTE EDIT (ON-LINE DATA LOGGING) *
4501     3998      . . .      ;*****
4502     3998      21 4D FF    LXI H,OUTDEV ;BOTH TAPES ON OUTPUT?
4503     3998      7E . .     MOV A,M
4504     399C      2F . .     CMA

```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 133
=====
4505     399D     E6  03  .      ANI  LFTCTU+RGCTU
4506     399F     3E  00  .      MVI  A,0          ;(SET A FOR "NO")
4507     39A1     C2  AA  39     JNZ  UTE050       ;NO -
4508     39A4     7E  .   .      MOV  A,M          ;YES - USE ONLY LEFT TAPE
4509     39A5     E6  FD  .      ANI  -1-RGTCTU
4510     39A7     77  .   .      MOV  M,A
4511     39A8     3E  FF  .      MVI  A,-1         ;AND SET SWPCTU = -1
4512     39AA     .   .   .      UTE050 EQU $
4513     39AA     05  .   .      DCK  B           ;SET B FOR BLINKING (LOGGING
4514     39AB     .   .   .      UTE070 EQU $
4515     39AB     32  24  FF     STA  SWPCTU      ;MARK SWPCTU ACCORDINGLY
4516     39AE     .   .   .      UTE100 EQU $
4517     39AE     3E  10  .      MVI  A,EDIT     ;FLAG FOR SET ROUTINE
4518     39B0     C3  0E  48     JMP  ZSTMD1
  
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 135
4570	39F3	CA	EB 39	JZ PTP010 ;YES - GET NEXT BYTE	
4571	39F6	FE	C4 .	CPI STPFLG ;NON-DISPLAYING TERMINATOR?	
4572	39F8	CA	EB 39	JZ PTP010 ;YES - GET NEXT BYTE	
4573	39F8	77	. .	MOV M,A ;PUT BYTE IN BUF	
4574	39FC	2C	. .	INR L ;INC POINTER	
4575	39FD	CA	4C 3A	JZ PTP490 ;REPORT BUFFER OVERFLOW	
4576	3A00	FE	CC .	CPI EOL ;WAS IT END OF LINE?	
4577	3A02	C2	EB 39	JNZ PTP010 ;NO - GET NEXT BYTE	
4578	3A05	2D	. .	DCR L ;YES - DELETE EOL FROM BUF	
4579	3A06	PTP020 EQU \$;FIND NEXT LINE POINTER	
4580	3A06	CD	91 00	CALL NXTCHR ;(NXTCHR RETURNS FILLS)	
4581	3A09	CA	06 3A	JZ PTP020 ;NOT THERE YET - CONTINUE	
4582	3A0C	*****	
4583	3A0C	; BUFFER FILLED, D,E -> NEXT LINE POINTER *	
4584	3A0C	*****	
4585	3A0C	PTP030 EQU \$	
4586	3A0C	1A	. . .	LDAX D ;GET NEXT BYTE	
4587	3A0D	D1	. . .	POP D ;D,E -> BUFFER STATUS	
4588	3A0E	D6	CE .	SUI EOP ;END OF PAGE?	
4589	3A10	C2	18 3A	JNZ PTP040 ;NO - OUTPUT LINE	
4590	3A13	85	. . .	ADD L ;YES - ANY BYTES IN BUF?	
4591	3A14	37	. . .	STC ;(PREPARE FOR ERROR EXIT)	
4592	3A15	CA	42 3A	JZ PTP220 ;NO - RELEASE BUF AND QUIT	
4593	3A18	PTP040 EQU \$;OUTPUT LINE	
4594	3A18	2D	. . .	DCR L ;IS LAST LINE A CR?	
4595	3A19	3E	0D .	MVI A,CR	
4596	3A1B	8E	. . .	CMP M	
4597	3A1C	CA	21 3A	JZ PTP050 ;YES - JUST APPEND LF	
4598	3A1F	2C	. . .	INR L ;NO - APPEND CR AND LF	
4599	3A20	77	. . .	MOV M,A	
4600	3A21	PTP050 EQU \$;APPEND LINE FEED	
4601	3A21	2C	. . .	INR L ;IS THERE ROOM FOR THE LF?	
4602	3A22	CA	4D 3A	JZ PTP500 ;NO - REPORT BUFFER OVERFLOW	
4603	3A25	36	0A .	MVI M,LF ;YES - INSERT IT	
4604	3A27	2C	. . .	INR L ;AND INCREMENT COUNT	
4605	3A28	*****	
4606	3A28	; (CR)LF APPENDED, MARK BUFFER FOR OUTPUT *	
4607	3A28	*****	
4608	3A28	1B	. . .	DCX D	
4609	3A29	1B	. . .	DCX D ;D,E -> LENGTH	
4610	3A2A	7D	. . .	MOV A,L ;L = LENGTH	
4611	3A2B	12	. . .	STAX D	
4612	3A2C	13	. . .	INX D ;D,E -> TYPE	
4613	3A2D	3E	FF .	MVI A,-1 ;-1 => DATA	
4614	3A2F	12	. . .	STAX D	
4615	3A30	13	. . .	INX D ;D,E -> STATUS	
4616	3A31	*****	
4617	3A31	; IF LOCAL READ IN PROGRESS, OUTDEV <- SAVOUT *	
4618	3A31	*****	
4619	3A31	CD	7A 41	CALL PUTIO ;OUTPUT BUFFER	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 136
=====
4620     3A34     D2  42  3A          JNC  PTP220      ;NO ERROR - QUIT
4621     3A37     3A  4F  FF          LDA  IOCERR      ;TIME TO SWAP CTU'S?
4622     3A3A     FE  7F  .           CPI  1770        ;(-1 => YES)
4623     3A3C     D4  97  3D          CNC  IOERCL      ;YES - CLEAR ERROR FLAG...
4624     3A3F     D4  7A  41          CNC  PUTIO       ;...AND TRY AGAIN
4625     3A42     .   .   .           PTP220 EQU $
4626     3A42     D1  .   .           POP  D           ;D,E -> BLOCKS FOR RELEASE
4627     3A43     3A  4F  FF          LDA  IOCERR      ;ANY ERRORS?
4628     3A46     FE  47  .           CPI  F+1         ;(U IS NOT A VALID OUTPT ERR
4629     3A48     D0  .   .           RNC              ;NO - RETURN NC => NO ERROR
4630     3A49     C3  84  36          JMP  USREXT      ;YES - RETURN C => CLR BUFS
4631     3A4C     .   .   .           ;*****
4632     3A4C     .   .   .           ; REPORT BUFFER OVERFLOW *
4633     3A4C     .   .   .           ;*****
4634     3A4C     .   .   .           PTP490 EQU $
4635     3A4C     D1  .   .           POP  D
4636     3A4D     .   .   .           PTP500 EQU $
4637     3A4D     2A  A0  00          LHLD BUFMSG     ;"BUFFER OVERFLOW" MESSAGE
4638     3A50     CD  B0  41          CALL IOFAI1     ;SET ERROR FLAG
4639     3A53     C3  42  3A          JMP  PTP220     ;REPORT ERROR
=====
  
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 137
4641	3A56	. . .	;	
4642	3A56	. . .	; * * * * *	
4643	3A56	. . .	;	
4644	3A56	. . .	; USREDA - PERFORM EDIT MODE "RECORD"	
4645	3A56	. . .	;	
4646	3A56	. . .	; LOCAL: COPY DISPLAY TO OUTPUT DEVICES,	
4647	3A56	. . .	; COPY FILE FROM INPUT TO OUTPUT,	
4648	3A56	. . .	; TERMINATE EDIT MODE.	
4649	3A56	. . .	;	
4650	3A56	. . .	; REMOTE: COPY DISPLAY TO OUTPUT DEVICES,	
4651	3A56	. . .	; STAY IN EDIT MODE.	
4652	3A56	. . .	;	
4653	3A56	. . .	; NO PARAMETERS, DESTROYS ALL REGISTERS	
4654	3A56	. . .	;	
4655	3A56	. . .	;	
4656	3A56	. . .	USREDA EQU \$	
4657	3A56	3A F3 FF	LOA MDFLG2 ;REMOTE OR LOCAL?	
4658	3A59	E6 08 .	ANI REMOTE	
4659	3A5B	CA 6E 3A	JZ EDA500 ;LOCAL -	
4660	3A5E	. . .	;*****	
4661	3A5E	. . .	; DO REMOTE "RECORD" *	
4662	3A5E	. . .	;*****	
4663	3A5E	. . .	EDA100 EQU \$	
4664	3A5E	CD 94 00	CALL GETDCM ;MONITOR DATACOM: DATA REC'D	
4665	3A61	CA 5E 3A	JZ EDA100 ;YES - CONTINUE MONITORING	
4666	3A64	CD 6A 00	CALL PTBLK0 ;WRITE TOP LINE TO OUT DEV'S	
4667	3A67	C2 5E 3A	JNZ EDA100 ;MORE TO DO - CONTINUE	
4668	3A6A	D8 . .	RC ;RETURN ON PTBLKS ERROR	
4669	3A6B	C3 9A 00	JMP MLKOFF ;FINISHED, NO ERRORS	
4670	3A6E	. . .	;*****	
4671	3A6E	. . .	; DO LOCAL "RECORD" *	
4672	3A6E	. . .	;*****	
4673	3A6E	. . .	EDA500 EQU \$	
4674	3A6E	CD 61 00	CALL CURPH ;LET USER SEE WHAT'S GOING O	
4675	3A71	. . .	EDA520 EQU \$;RECORD SCREEN	
4676	3A71	3E 1A .	MVI A,26 ;PUT CURSOR OFF SCREEN TO	
4677	3A73	32 C0 FF	STA CURROW ;AVOID PTBLK ERRORS	
4678	3A76	CD 2E 48	CALL RETSCN ;USER INTERRUPT?	
4679	3A79	DA 61 00	JC CURPH ;YES - HOME UP AND RETURN	
4680	3A7C	CD 6A 00	CALL PTBLK0 ;WRITE TOP LINE TO OUT DEV'S	
4681	3A7F	C2 71 3A	JNZ EDA520 ;MORE LINES - CONTINUE	
4682	3A82	. . .	;*****	
4683	3A82	. . .	; ONLY ONE LINE LEFT ON DISPLAY *	
4684	3A82	. . .	;*****	
4685	3A82	D8 . .	RC ;QUIT ON PTBLK ERROR	
4686	3A83	CD 9A 00	CALL MLKOFF ;TURN OFF MEMORY LOCK	
4687	3A86	CD 97 00	CALL MLKSCH ;GET LAST ROW (IF ANY)	
4688	3A89	CA 95 3A	JZ EDA550 ;NO UNLOCKED ROW	
4689	3A8C	2B . .	DCX H ;UNLOCKED ROW - GET 1ST CHAR	
4690	3A8D	7E . .	MOV A,M	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
4691     3A8E     23      .      .      INX  H
4692     3A8F     FE      CC      .      CPI  EOL      ;IS IT A NULL ROW?
4693     3A91     EB      .      .      XCHG                ;PTTPLN WANTS D,E -> LINE
4694     3A92     C4      C0      39      CNZ  PTTPLN      ;NO - OUTPUT IT
4695     3A95     .      .      .      EDA550 EQU $
4696     3A95     CD      61      00      CALL CURPH      ;HOME CURSOR
4697     3A98     CD      6D      00      CALL CLEARL     ;DELETE TOP LINE (IF ANY)
4698     3A9B     .      .      .      ;*****
4699     3A9B     .      .      .      ; COPY REST OF INPUT TO OUTPUT *
4700     3A9B     .      .      .      ;*****
4701     3A9B     3A      4E      FF      LDA  INPDEV     ;GET INPUT DEVICE
4702     3A9E     FE      04      .      CPI  DISPLY     ;IS IT DISPLAY?
4703     3AA0     CA      5B      39      JZ   USRTED     ;YES - TURN OFF EDIT & RETUR
4704     3AA3     E6      03      .      ANI  LFTCTU+RGCTU
4705     3AA5     CA      C2      3A      JZ   EDA600     ;INPUT NOT TAPE, CONTINUE
4706     3AA8     21      66      FF      LXI  H,CTSTAT  ;IS INPUT TAPE INSERTED?
4707     3AAB     A6      .      .      ANA  M
4708     3AAC     CA      5B      39      JZ   USRTED     ;NO - QUIT
4709     3AAF     67      .      .      MOV  H,A       ;SAVE INPUT UNIT
4710     3AB0     CD      DC      2A      CALL GTCTBT     ;GET BIT FOR SEL UNIT
4711     3AB3     BC      .      .      CMP  H         ;IS INPUT UNIT SELECTED?
4712     3AB4     3A      62      FF      LDA  CNTRL0     ;YES - GET BITS FOR SEL. UNI
4713     3AB7     CA      BD      3A      JZ   EDA580
4714     3ABA     3A      5B      FF      LDA  CNTRL0+OTHER-SFTCNT ;NO - GET OTHER
4715     3ABD     .      .      .      EDA580 EQU $
4716     3ABD     E6      01      .      ANI  EOF       ;TAPE AT EOF?
4717     3ABF     C2      5B      39      JNZ  USRTED     ;YES - QUIT
4718     3AC2     .      .      .      EDA600 EQU $
4719     3AC2     06      00      .      MVI  B,0       ;(ENTRY FOR OTHER DEVICES)
4720     3AC4     CD      CA      3E      CALL XFRD2D     ;(SET FOR XFR TO EOF)
4721     3AC7     D4      5B      39      CNC  USRTED     ;NO - TRANSFER TO END OF FIL
4722     3ACA     C3      84      36      JMP  USREXT     ;NO ERRORS - TURN OFF EDIT
                                ;REPORT ANY ERRORS & RET
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
4724	3ACD	.	.	;
4725	3ACD	.	.	;
4726	3ACD	.	.	; USRXFL - USER TRANSFER ONE RECORD
4727	3ACD	.	.	;
4728	3ACD	.	.	USRXFL EQU \$
4729	3ACD	05	.	DCR B
4730	3ACE	.	.	;
4731	3ACE	.	.	;
4732	3ACE	.	.	; USRXFF - USER TRANSFER TO END OF FILE MARK
4733	3ACE	.	.	;
4734	3ACE	.	.	USRXFF EQU \$
4735	3ACE	05	.	DCR B
4736	3ACF	.	.	;
4737	3ACF	.	.	;
4738	3ACF	.	.	; DCR BUSER TRANSFER TO END OF FI
4739	3ACF	.	.	;
4740	3ACF	.	.	USRXFA EQU \$
4741	3ACF	.	.	; FALL THROUGH TO USRXFR
4742	3ACF	.	.	;
4743	3ACF	.	.	; * * * * *
4744	3ACF	.	.	;
4745	3ACF	.	.	; USRXFR - USER INITIATED TRANSFER
4746	3ACF	.	.	;
4747	3ACF	.	.	; ENTRY: B = TRANSFER LIMIT
4748	3ACF	.	.	;
4749	3ACF	.	.	USRXFR EQU \$
4750	3ACF	B7	.	ORA A ;NC => TRANSFER
4751	3AD0	C3	CA 3E	JMP XFRD2D ;PERFORM TRANSFER

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 140
4753	3AD3	.	.	. ;*****	
4754	3AD3	.	.	. ; DISPLAY MESSAGES *	
4755	3AD3	.	.	. ;*****	
4756	3AD3	.	.	. ;	
4757	3AD3	.	.	. MESSGE EQU \$;FOR CROSS REFERENCE	
4758	3AD3	.	.	. ;	
4759	3AD3	.	.	. SKPMSG EQU \$	
4760	3AD3	82	53	4B DB INVRS,'SKIP LINES',0	
4761	3ADF	.	.	. LLPMSG EQU \$	
4762	3ADF	82	4C	4F DB INVRS,'LOCATING LOAD POINT',0	
4763	3AF4	.	.	. OFFMSG EQU \$	
4764	3AF4	83	52	55 DB 203Q,'RUNOFF',0	
4765	3AFC	.	.	. UETMSG EQU \$	
4766	3AFC	83	41	42 DB 203Q,'ABORTED - '	
4767	3B07	.	.	. EOTMSG EQU \$	
4768	3B07	82	45	4E DB INVRS,'END OF TAPE',0	
4769	3B14	.	.	. FEVDMS EQU \$	
4770	3B14	82	46	49 DB INVRS,'FIND '	
4771	3B1A	.	.	. EVDMSG EQU \$	
4772	3B1A	82	45	4E DB INVRS,'END OF DATA',0	
4773	3B27	.	.	. ;	
4774	3B27	.	.	. NRCMSG EQU \$	
4775	3B27	82	44	41 DB INVRS,'DATA PROTECTED',0	
4776	3B37	.	.	. ;	
4777	3B37	.	.	. LOCMSG EQU \$	
4778	3B37	82	46	49 DB INVRS,'FIND FILE',0	
4779	3B42	.	.	. ;	
4780	3B42	.	.	. FLINMS EQU \$	
4781	3B42	82	46	49 DB INVRS,'FILE NUMBER',200Q,' '	
4782	3B50	8A	49	4E DB 212Q,'INCHES LEFT',200Q,' ': ',INVRS,0	
4783	3B62	.	.	. ;	
4784	3B62	.	.	. HRDMSG EQU \$	
4785	3B62	82	52	45 DB 202Q,'READ '	
4786	3B68	.	.	. FAILMS EQU \$;GENERAL FAIL MESSAGE	
4787	3B68	82	46	41 DB INVRS,'FAIL',0	
4788	3B6E	.	.	. NTPMSG EQU \$	
4789	3B6E	82	4E	4F DB INVRS,'NO TAPE',0	
4790	3B77	.	.	. INOMSG EQU \$	
4791	3B77	82	22	46 DB 202Q,'"FROM" DEVICE = "TO" DEVICE'	
4792	3B93	CE	.	. DB EOP	
4793	3B94	.	.	. TMTMSG EQU \$	
4794	3B94	82	54	4F DB 202Q,'TOO MANY "TO" DEVICES',EOP	
4795	3BAB	.	.	. PREMSG EQU \$	
4796	3BAB	82	50	52 DB INVRS,'PRINT FAIL',EOP	

=====					PAGE 141		=====				
ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS						
=====											
4798	3BB7	.	.	.	RTRYMS	EQU	\$				
4799	3BB7	82	52	45	DB	202Q,	'RETRY'				
4800	3BBD	.	.	.	EOPMSG	EQU	\$				
4801	3BBD	CE	.	.	DB	EOP					
4802	3BBE	80	.	.	NLTPMS	DB	200Q				
4803	3BBF	.	.	.	OLTPMS	EQU	\$				
4804	3BBF	20	4F	4E	DB	' ON LEFT DRIVE',	0				
4805	3BCE	80	.	.	N RTPMS	DB	200Q				
4806	3BCF	.	.	.	ORTPMS	EQU	\$				
4807	3BCF	20	4F	4E	DB	' ON RIGHT DRIVE',	0				
4808	3BDF	.	.	.	TMFMSG	EQU	\$				
4809	3BDF	82	54	4F	DB	INVR,	'TOO MANY "FROM" DEVICES',	EOP			
4810	3BF8	.	.	.	BSYMSG	EQU	\$				
4811	3BF8	82	42	55	DB	INVR,	'BUSY - WAITING',	0			
4812	3C08	.	.	.	WRFMSG	EQU	\$				
4813	3C08	82	57	52	DB	INVR,	'WRITE FAIL',	0			
4814	3C14	.	.	.	NULMSG	EQU	\$				
4815	3C14	00	.	.	DB	0					
4816	3C15	.	.	.	EOFMSG	EQU	\$				
4817	3C15	82	45	4E	DB	INVR,	'END OF FILE',	0			
4818	3C22	.	.	.	DLRMSG	EQU	\$				
4819	3C22	82	44	49	DB	202Q,	'DIFFERENT LENGTH RECORDS',	0			
4820	3C3C	.	.	.	DIFMSG	EQU	\$				
4821	3C3C	82	44	49	DB	202Q,	'DIFFERENCE IN BYTE ',	0			
4822	3C51	.	.	.	RECMMSG	EQU	\$				
4823	3C51	2C	20	52	DB	', RECORD ',	0				
4824	3C5B	.	.	.	FILMSG	EQU	\$				
4825	3C5B	2C	20	46	DB	', FILE'					
4826	3C61	.	.	.	BLKMSG	EQU	\$				
4827	3C61	20	00	.	DB	' ',	0				
4828	3C63	.	.	.	STALMS	EQU	\$				
4829	3C63	83	53	54	DB	203Q,	'STALL',	0			
4830	3C6A	.	.	.	FMSMSG	EQU	\$				
4831	3C6A	82	46	49	DB	INVR,	'FILE MISSING',	0			
4832	3C78	.	.	.	CONMSG	EQU	\$				
4833	3C78	82	43	4F	DB	INVR,	'CONFLICTING I/O',	EOP			
4834	3C89	.	.	.							
4835	3C89	.	.	.	M1MSG	EQU	\$				
4836	3C89	20	20	31	DB	' -1',	EOP				
4837	3C8D	.	.	.							
4838	3C8D	.	.	.	PLSMSG	EQU	\$				
4839	3C8D	20	28	00	DB	' +',	0				
4840	3C90	.	.	.							
4841	3C90	.	.	.	MNSMSG	EQU	\$				
4842	3C90	20	20	00	DB	' -',	0				
4843	3C93	.	.	.							
4844	3C93	.	.	.	COMAMS	EQU	\$				
4845	3C93	2C	20	20	DB	', ',	INVR,	0			

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 142
=====
4847      3C98      . . .      ;
4848      3C98      . . .      ; DSPNUM - CONVERT NUMBER AND PLACE IN MEMORY
4849      3C98      . . .      ;
4850      3C98      . . .      ;          ENTRY: (H,L) = ADDRESS FOR CHARS.
4851      3C98      . . .      ;          A = NUMBER TO BE CONVERTED
4852      3C98      . . .      ;
4853      3C98      . . .      ;          EXIT: (H,L) = ADDRESS AFTER NUMBER
4854      3C98      . . .      ;
4855      3C98      . . .      DSPNUM EQU $
4856      3C98      C5 . . .      PUSH B          ;SAVE REGISTER B-E
4857      3C99      D5 . . .      PUSH D
4858      3C9A      CD AB 00      CALL BN2DE0
4859      3C9D      36 00 .      MVI M,0         ;ADD NULL TO CONTINUE MESSAG
4860      3C9F      23 . . .      INX H          ;SET TO NEXT MESSAGE BYTE
4861      3CA0      D1 . . .      POP D          ;RESTORE REGISTER B-E
4862      3CA1      C1 . . .      POP B
4863      3CA2      C9 . . .      RET           ;RETURN
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 143
4865	3CA3	.	.	;	
4866	3CA3	.	.	;	CARDIO - DISPLAY FIRST HALF OF MESSAGE ONLY
4867	3CA3	.	.	;	
4868	3CA3	.	.	;	ENTRY: H,L = POINTER TO MESSAGE
4869	3CA3	.	.	;	
4870	3CA3	.	.	;	CARDIO EQU \$
4871	3CA3	CD	F0 2D	CALL SLTPMS	;GET MSG FOR SELECTED TAPE
4872	3CA6	.	.	;	
4873	3CA6	.	.	;	*****
4874	3CA6	.	.	;	
4875	3CA6	.	.	;	CARDIS - SET DISPLAY TO I/O MESSAGE
4876	3CA6	.	.	;	
4877	3CA6	.	.	;	ENTRY: MSGPT1 = PTR TO 1ST HALF OF MSG
4878	3CA6	.	.	;	MSGPT3. . .MSGPT8 = PTRS TO REST OF
4879	3CA6	.	.	;	MESSAGE
4880	3CA6	.	.	;	
4881	3CA6	.	.	;	EXIT : A,H,L DESTROYED
4882	3CA6	.	.	;	H = BASEH
4883	3CA6	.	.	;	
4884	3CA6	.	.	;	ONE TO EIGHT MESSAGE POINTERS MAY BE USED.
4885	3CA6	.	.	;	ALL PARTS END IN '0' EXCEPT THE LAST, WHICH END
4886	3CA6	.	.	;	IN EOP.
4887	3CA6	.	.	;	
4888	3CA6	.	.	;	CARDIS EQU \$
4889	3CA6	37	.	STC	;C => CLOBBER DISPLAY FIRST
4890	3CA7	C5	.	PUSH B	
4891	3CA8	D5	.	PUSH D	
4892	3CA9	CD	40 00	CALL DSPMSG	;DISPLAY MESSAGE
4893	3CAC	D1	.	POP D	
4894	3CAD	C1	.	POP B	
4895	3CAE	C9	.	RET	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 144
=====
4897      3CAF      . . .      ;*****
4898      3CAF      . . .      ; ESCAPE SEQUENCE CONTROL *****
4899      3CAF      . . .      ;*****
4900      3CAF      . . .      ;*****
4901      3CAF      . . .      ; BINARY TRANSFER *
4902      3CAF      . . .      ;*****
4903      3CAF      . . .      CTDCDP EQU $
4904      3CAF      CD 97 3D      CALL IOERCL      ;CLEAR I/O ERROR FLAG
4905      3CB2      3E 0A .      MVI A,FSTBIN    ;SET DATACOM FOR FAST (9600)
4906      3CB4      CD F4 3C      CALL DCMCTL      ;BINARY
4907      3CB7      CD C1 2D      CALL BSYCK0     ;INPUT TAPE => WAIT TILL FRE
4908      3CBA      D8 . .      RC              ;RETURN ON USER INTERRUPT
4909      3CBB      3E F8 .      MVI A,-1-FILRED-USREAD-RDWOWT
4910      3CBD      CD 2A 2B      CALL CLIQFS     ;CLEAR USER & FILE READ FLAG
4911      3CC0      C4 75 43      CNZ RDABR1     ;IF NOT ZERO, ABORT READ
4912      3CC3      CD A3 44      CALL INTDS0    ;INPUT=DISPLAY => INITIALIZE
4913      3CC6      . . .      CTD050 EQU $
4914      3CC6      CD 64 41      CALL GETIO     ;GET A RECORD
4915      3CC9      06 00 .      MVI B,0        ;(SET B-REG FOR EXIT)
4916      3CCB      DA E3 3C      JC CTD100     ;EXIT ON INPUT ERROR
4917      3CCE      3E 20 .      MVI A,DATCOM  ;MARK FOR OUTPUT TO DATACOM
4918      3CD0      12 . .      STAX D
4919      3CD1      1B . .      DCX D          ;GET RECORD TYPE
4920      3CD2      1A . .      LDAX D
4921      3CD3      B7 . .      ORA A         ;DATA RECORD?
4922      3CD4      F2 E4 3C      JP CTD110     ;NO - EXIT
4923      3CD7      1B . .      DCX D          ;BNR010 WANTS D,E->LENGTH
4924      3CD8      CD 2A 3D      CALL GETPTR    ;AND H,L -> FIRST BYTE
4925      3CDB      CD 2A 43      CALL BNR010    ;SEND THE RECORD
4926      3CDE      06 00 .      MVI B,0        ;(SET B-REG FOR EXIT)
4927      3CE0      D2 C6 3C      JNC CTD050    ;NO ERROR - GET NEXT RECORD
4928      3CE3      . . .      ;*****
4929      3CE3      . . .      ; I/O COMPLETED - SEND TWO-BYTE RECORD: *
4930      3CE3      . . .      ; 0, 0 => SUCCESS *
4931      3CE3      . . .      ; -1,-1 => FAILURE *
4932      3CE3      . . .      ; RESTORE BAUD RATE AND RETURN *
4933      3CE3      . . .      ;*****
4934      3CE3      . . .      CTD100 EQU $   ;ERROR RETURN
4935      3CE3      05 . .      DCR B
4936      3CE4      . . .      CTD110 EQU $   ;SUCCESSFUL COMPLETION
4937      3CE4      78 . .      MOV A,B
4938      3CE5      CD 7C 00      CALL XPUTD3    ;OUTPUT CHAR
4939      3CE8      78 . .      MOV A,B
4940      3CE9      CD 7C 00      CALL XPUTD3    ;OUTPUT SECOND CHAR
4941      3CEC      CD 17 3D      CALL FREBFS    ;CLEAR BOTH BUFFERS
4942      3CEF      CD 23 50      CALL ZNDBIN    ;RESET BAUD, GO TO ASCII
4943      3CF2      . . .      ; SIGNAL END OF DATA BLOCK
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
4946	3CF2	.	.	;
4947	3CF2	.	.	; * * * * *
4948	3CF2	.	.	;
4949	3CF2	.	.	; ENDATA - SIGNAL END OF DATA BLOCK
4950	3CF2	.	.	;
4951	3CF2	.	.	;
4952	3CF2	.	.	ENDATA EQU \$
4953	3CF2	3E	07	MVI A,ENDBLK
4954	3CF4	.	.	;
4955	3CF4	.	.	; * * * * *
4956	3CF4	.	.	;
4957	3CF4	.	.	; DCMCTL - PERFORM DATACOM CONTROL FUNCTION
4958	3CF4	.	.	;
4959	3CF4	.	.	;
4960	3CF4	.	.	DCMCTL EQU \$
4961	3CF4	CD	11 50	CALL ZDCCTL ;PERFORM CONTROL
4962	3CF7	D0	.	RNC ;RETURN IF NO ERROR
4963	3CF8	CA	14 48	JZ ZBELL ;RING AND RETURN IF NU MSG
4964	3CFB	.	.	;
4965	3CFB	.	.	; * * * * *
4966	3CFB	.	.	;
4967	3CFB	.	.	; DCERR - HANDLE DATACOM ERRORS
4968	3CFB	.	.	;
4969	3CFB	.	.	ENTRY: C,Z => NON-FATAL
4970	3CFB	.	.	C,NZ => FATAL
4971	3CFB	.	.	H,L -> ERROR MESSAGE
4972	3CFB	.	.	;
4973	3CFB	.	.	EXIT : NON-FATAL - JUMP TO IOFAIL
4974	3CFB	.	.	FATAL - DISPLAY MESSAGE AND HANG
4975	3CFB	.	.	;
4976	3CFB	.	.	DCERR EQU \$
4977	3CFB	C8	.	RZ ;NON-FATAL - EXIT
4978	3CFC	C3	9D 00	JMP HANGUO ;DISPLAY MSG AND QUIT

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
4980	3CFF	.	.	.	;	
4981	3CFF	.	.	.	; * * * * *	
4982	3CFF	.	.	.	;	
4983	3CFF	.	.	.	; GTIOB0 - WAIT UNTIL A BUFFER IS FREE	
4984	3CFF	.	.	.	;	
4985	3CFF	.	.	.	; ENTRY: DON'T CARE	
4986	3CFF	.	.	.	;	
4987	3CFF	.	.	.	; EXIT : C => CTU ERROR OR USER INTERRUPT	
4988	3CFF	.	.	.	; NC => BUFFER FOUND	
4989	3CFF	.	.	.	; H,L -> STATUS	
4990	3CFF	.	.	.	; A,H,L DESTROYED	
4991	3CFF	.	.	.	;	
4992	3CFF	.	.	.	; GTIOB0 EQU \$	
4993	3CFF	CD	C1	29	CALL CTMON1 ;MONITOR TAPES (C,Z => ERROR	
4994	3D02	D4	31	48	CNC RETSCO ;MONITOR KYBD (C,Z => USER	
4995	3D05	.	.	.	; INTERRUPT)	
4996	3D05	D4	0C	3D	CNC GTIOBF ;BUFFER FREE? (Z => YES)	
4997	3D08	C8	.	.	RZ ;ERROR OR FREE BUFFER	
4998	3D09	C3	FF	3C	JMP GTIOB0 ;CONTINUE WAITING	
4999	3D0C	.	.	.	;	
5000	3D0C	.	.	.	; * * * * *	
5001	3D0C	.	.	.	;	
5002	3D0C	.	.	.	; GTIOBF - FIND AN EMPTY I/O BUFFER	
5003	3D0C	.	.	.	;	
5004	3D0C	.	.	.	; ENTRY: DON'T CARE	
5005	3D0C	.	.	.	;	
5006	3D0C	.	.	.	; EXIT : NC	
5007	3D0C	.	.	.	; Z => SUCCESS	
5008	3D0C	.	.	.	; H,L -> STATUS	
5009	3D0C	.	.	.	; NZ => BOTH BUFFERS BUSY	
5010	3D0C	.	.	.	; H,L DESTROYED	
5011	3D0C	.	.	.	; A DESTROYED	
5012	3D0C	.	.	.	;	
5013	3D0C	.	.	.	; GTIOBF EQU \$	
5014	3D0C	21	3A	FF	LXI H,B1STAT ;BUF 1 AVAILABLE?	
5015	3D0F	7E	.	.	MOV A,M	
5016	3D10	B7	.	.	ORA A	
5017	3D11	C8	.	.	RZ ;YES - RETURN	
5018	3D12	2E	37	.	MVI L,B2STAT*256/256	
5019	3D14	7E	.	.	MOV A,M ;BUF 2 AVAILABLE?	
5020	3D15	B7	.	.	ORA A	
5021	3D16	C9	.	.	RET	
5022	3D17	.	.	.	;	
5023	3D17	.	.	.	; * * * * *	
5024	3D17	.	.	.	;	
5025	3D17	.	.	.	; FREBFS - FREE BOTH I/O BUFFERS	
5026	3D17	.	.	.	;	
5027	3D17	.	.	.	; ENTRY: DON'T CARE	
5028	3D17	.	.	.	;	
5029	3D17	.	.	.	; EXIT : A = 0	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                PAGE 147
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
5030	3D17	.	.	;
5031	3D17	.	.	FREBFS EQU \$
5032	3D17	97	.	SUB A
5033	3D18	32	3A	FF STA B1STAT
5034	3D1B	32	37	FF STA B2STAT
5035	3D1E	C9	.	RET

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 148
=====
5037      3D1F      . . .      ;
5038      3D1F      . . .      ; * * * * *
5039      3D1F      . . .      ;
5040      3D1F      . . .      ;      CHGBUF - LOOK AT OTHER I/O BUFFER
5041      3D1F      . . .      ;
5042      3D1F      . . .      ;      ENTRY:  D,E -> BXSTAT,TYPE,OR LEN
5043      3D1F      . . .      ;
5044      3D1F      . . .      ;      EXIT :  D,E -> STAT OF OTHER BUFFER
5045      3D1F      . . .      ;      A DESTROYED
5046      3D1F      . . .      ;
5047      3D1F      . . .      ;
5048      3D1F      . . .      CHGBUF EQU $      ;LOOK AT OPPOSITE BUFFER
5049      3D1F      7B . .      MOV A,E
5050      3D20      FE 38 .      CPI B1LEN*256/256
5051      3D22      11 3A FF      LXI D,B1STAT
5052      3D25      D8 . .      RC
5053      3D26      11 37 FF      LXI D,B2STAT
5054      3D29      C9 . .      RET
5055      3D2A      . . .      ;
5056      3D2A      . . .      ; * * * * *
5057      3D2A      . . .      ;
5058      3D2A      . . .      ;      GETPTR - GET POINTER TO 1ST BYTE OF I/O BUF
5059      3D2A      . . .      ;
5060      3D2A      . . .      ;      ENTRY:  D,E -> I/O BUFFER STATUS, TYPE, LENGT
5061      3D2A      . . .      ;
5062      3D2A      . . .      ;      ENTRY GETPT1: A = LOW BYTE OF POINTER
5063      3D2A      . . .      ;
5064      3D2A      . . .      ;      EXIT :  H,L -> FIRST BYTE OF ASSOCIATED BUF
5065      3D2A      . . .      ;
5066      3D2A      . . .      ;
5067      3D2A      . . .      GETPTR EQU $
5068      3D2A      7B . .      MOV A,E
5069      3D28      . . .      GETPT1 EQU $
5070      3D28      FE 38 .      CPI B1LEN*256/256
5071      3D2D      21 00 FC      LXI H,IOBUF1
5072      3D30      D0 . .      RNC
5073      3D31      21 00 FD      LXI H,IOBUF2
5074      3D34      C9 . .      RET
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 149
5076	3D35	.	.	.	;
5077	3D35	.	.	.	; I/O ENTRY TABLES
5078	3D35	.	.	.	;
5079	3D32	.	.	.	IOCTAB EQU \$-3
5080	3D35	.	.	.	;
5081	3D35	20	20	.	DB 40Q,40Q ;IGNORE BLANKS IN ESCAPE
5082	3D37	93	8D	.	DW IOC010+B15 ;SEQUENCE
5083	3D39	.	.	.	;
5084	3D39	28	28	.	DB 53Q,53Q ;+ - SET SIGN FLAG TO +1
5085	3D3B	49	80	.	DW DCPLUS+B15
5086	3D3D	2D	2D	.	DB 55Q,55Q ;- - SET SIGN FLAG TO -1
5087	3D3F	4C	80	.	DW DCMNUS+B15
5088	3D41	.	.	.	;
5089	3D41	30	39	.	DB 60Q,71Q ;DIGITS <0> TO <9>
5090	3D43	46	80	.	DW DCNUM+B15 ;ACCUMULATE DECIMAL VALUE
5091	3D45	.	.	.	;
5092	3D45	42	46	.	DB 102Q,106Q ;RANGE FROM TO <F>
5093	3D47	61	3D	.	DW IOCT20
5094	3D49	62	66	.	DB 142Q,146Q ;LOWER CASE CHAR RANGE
5095	3D4B	61	3D	.	DW IOCT20
5096	3D4D	.	.	.	;
5097	3D4D	4D	57	.	DB 115Q,127Q ;RANGE FROM <M> TO <W>
5098	3D4F	6B	3D	.	DW IOCT30
5099	3D51	6D	77	.	DB 155Q,167Q ;LOWER CASE CHAR RANGE
5100	3D53	6B	3D	.	DW IOCT30
5101	3D55	.	.	.	;
5102	3D55	5E	5E	.	DB 136Q,136Q ;CHARACTER <^>
5103	3D57	22	8E	.	DW IOC110+B15 ;PROCESS STATUS REQUEST
5104	3D59	7E	7E	.	DB 176Q,176Q ;LOWER CASE <^>
5105	3D5B	22	8E	.	DW IOC110+B15
5106	3D5D	.	.	.	;
5107	3D5D	.	.	.	;*****
5108	3D5D	00	7F	.	DB 0Q,177Q ;CATCH ALL ROUTINE
5109	3D5F	.	.	.	;*****
5110	3D5F	4F	80	.	DW ESCEND+B15 ;GO TO ERROR ROUTINE

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 150
=====
5112     3D61      . . .      ;
5113     3D61      . . .      ; ENTRY LISTS
5114     3D61      . . .      ;
5115     3D61      . . .      IOCT20 EQU $
5116     3D61     AD 3D      DW   IOC020 ;B - SINGLE RECORD TRANSFER
5117     3D63     BD 3D      DW   IOC030 ;C - CONTROL FUNCTION
5118     3D65     0B 3E      DW   IOC080 ;D - DESTINATION DEVICE
5119     3D67     4F 00      DW   ESCEND ;E - INVALID, FLAG ERROR
5120     3D69     AC 3D      DW   IOC050 ;F - SINGLE FILE TRANSFER
5121     3D6B      . . .      ;
5122     3D6B      . . .      IOCT30 EQU $
5123     3D6B     AB 3D      DW   IOC070 ;M - TRANSFER TO END OF DATA
5124     3D6D     4F 00      DW   ESCEND ;N - INVALID, FLAG ERROR
5125     3D6F     4F 00      DW   ESCEND ;O - INVALID, FLAG ERROR
5126     3D71     F2 3D      DW   IOC090 ;P - CONTROL PARAMETER
5127     3D73     4F 00      DW   ESCEND ;Q - INVALID, FLAG ERROR
5128     3D75     BB 3D      DW   IOC100 ;R - READ RECORD TO DATACOM
5129     3D77     0C 3E      DW   IOC060 ;S - DEFINE SOURCE DEVICE
5130     3D79     4F 00      DW   ESCEND ;T - INVALID, FLAG ERROR
5131     3D7B     01 3E      DW   IOC040 ;U - UNIT SPECIFIER
5132     3D7D     4F 00      DW   ESCEND ;V - INVALID, FLAG ERROR
5133     3D7F     D6 3D      DW   IOC120 ;W - WRITE DATA FROM DATACOM
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 151
5135	3D81	. . .	;	
5136	3D81	. . .	; * * * * *	
5137	3D81	. . .	;	
5138	3D81	. . .	; IOCNTL - I/O CONTROL	
5139	3D81	. . .	;	
5140	3D81	. . .	; <ESC><&><LOWER CASE P> . . .	
5141	3D81	. . .	;	
5142	3D81	. . .	IOCNTL EQU \$	
5143	3D81	21 32 3D	LXI H,IOCTAB ;SET ESCAPE SEQUENCE	
5144	3D84	22 D2 FF	SHLD RNGTA ;RANGE TABLE POINTER	
5145	3D87	. . .	IOCCLR EQU \$	
5146	3D87	21 DF FF	LXI H,IODATA+1 ;SET STARTING ADDRESS	
5147	3D8A	0E 0B .	MVI C,11 ;SET NUMBER OF BYTES	
5148	3D8C	. . .	;	
5149	3D8C	. . .	; IOCCL1 - CLEAR RAM AREA	
5150	3D8C	. . .	;	
5151	3D8C	. . .	; ENTRY: C = NUMBER OF BYTES TO BE CLEARED	
5152	3D8C	. . .	; H = BASEH	
5153	3D8C	. . .	; L = UPPER ADDRESS LIMIT	
5154	3D8C	. . .	;	
5155	3D8C	. . .	; EXIT : H,L = ESCFLG	
5156	3D8C	. . .	;	
5157	3D8C	. . .	IOCCL1 EQU \$	
5158	3D8C	AF . . .	XRA A ;CLEAR TO ZERO	
5159	3D8D	. . .	;	
5160	3D8D	. . .	IOC002 EQU \$	
5161	3D8D	77 . . .	MOV M,A	
5162	3D8E	2D . . .	DCR L ;DECREMENT ADDRESS	
5163	3D8F	0D . . .	DCR C ;ALL LOCATIONS DONE?	
5164	3D90	C2 8D 3D	JNZ IOC002 ;NO - DO NEXT BYTE	
5165	3D93	. . .	IOC010 EQU \$	
5166	3D93	2E D1 .	MVI L,ESCFLG	
5167	3D95	36 02 .	MVI M,2	
5168	3D97	. . .	;	
5169	3D97	. . .	; IOERCL - CLEAR I/O ERROR FLAG	
5170	3D97	. . .	; IOCERR=S => SUCCESS	
5171	3D97	. . .	;	
5172	3D97	. . .	IOERCL EQU \$	
5173	3D97	3E 53 .	MVI A,S	
5174	3D99	32 4F FF	STA IOCERR	
5175	3D9C	C9 . . .	RET	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 152
=====
5177      3D9D      . . .      ;*****
5178      3D9D      . . .      ; I/O COMMAND TABLE AND EQUATES *
5179      3D9D      . . .      ;*****
5180      3D9B      . . .      IOCMTB EQU $-2
5181      3D9D      C4 3E .      DW XFRREC ;1 - TRANSFER/COMPARE RECORD
5182      3D9F      C5 3E .      DW XFEOF ;2 - TRANSFER/COMPARE A FILE
5183      3DA1      C6 3E .      DW XFREVD ;3 - TRANSFER/COMPARE MEDIUM
5184      3DA3      98 3F .      DW CTRLIO ;4 - I/O CONTROL
5185      3DA5      AB 40 .      DW IOWRIT ;5 - STORE DATACOM RECORD
5186      3DA7      67 40 .      DW IOREAD ;6 - READ RECORD TO DATA COM
5187      3DA9      92 40 .      DW IOSTAT ;7 - GET I/O STATUS
5188      3DAB      . . .      ;
5189      0001      . . .      IOBNUM EQU 1 ;TRANSFER/COMPARE A RECORD
5190      0002      . . .      IOFNUM EQU 2 ;TRANSFER/COMPARE A FILE
5191      0003      . . .      IOMNUM EQU 3 ;TRANSFER/COMPARE MEDIUM
5192      0004      . . .      IOCNUM EQU 4 ;EXECUTE I/O CONTROL
5193      0005      . . .      IOWNUM EQU 5 ;STORE RECORD FROM DATA COMM
5194      0006      . . .      IORNUM EQU 6 ;READ RECORD TO DATA COMM
5195      0007      . . .      IOSNUM EQU 7 ;GET I/O STATUS
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 153
=====
5197      3DAB      . . .      ;
5198      3DAB      . . .      ; <M> - TRANSFER TO END OF DATA
5199      3DAB      . . .      ;
5200      3DAB      . . .      IOC070 EQU $
5201      3DAB      04 . .      INR B          ;COMMAND CODE = 3
5202      3DAC      . . .      ;
5203      3DAC      . . .      ; <F> - SINGLE FILE TRANSFER
5204      3DAC      . . .      ;
5205      3DAC      . . .      IOC050 EQU $
5206      3DAC      04 . .      INR B          ;COMMAND CODE = 2
5207      3DAD      . . .      ;
5208      3DAD      . . .      ; <B> - SINGLE RECORD TRANSFER
5209      3DAD      . . .      ;
5210      3DAD      . . .      IOC020 EQU $
5211      3DAD      . . .      ;          COMMAND CODE = 1
5212      3DAD      . . .      ;
5213      3DAD      CD AA 3E      CALL CHKCMD ;ANY PREVIOUS COMMAND?
5214      3DB0      C0 . .      RNZ          ;YES - ABORT ESCAPE SEQUENCE
5215      3DB1      70 . .      MOV M,B      ;NO - STORE COMMAND CODE
5216      3DB2      3A DE FF      LDA IODATA   ;SAVE COMMAND MODIFIER FOR
5217      3DB5      32 D8 FF      STA IOCTYP   ;LATER ANALYSIS
5218      3DB8      C3 34 3E      JMP IOCEX0   ;CLEAR PARM VALUE AND EXIT
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
5220      3DBB      . . .      ;
5221      3DBB      . . .      ; <R> - READ A RECORD TO THE DATACOM
5222      3DBB      . . .      ;
5223      3DBB      . . .      IOC100 EQU $
5224      3DBB      04 . .      INR B          ;COMMAND CODE = 6
5225      3DBC      04 . .      INR B
5226      3DBD      . . .      ;
5227      3DBD      . . .      ; <C> - CONTRUL FUNCTION
5228      3DBD      . . .      ;
5229      3DBD      . . .      IOC030 EQU $
5230      3DBD      04 . .      INR B          ;COMMAND CODE = 4
5231      3DBE      04 . .      INR B
5232      3DBF      04 . .      INR B
5233      3DC0      . . .      ;
5234      3DC0      CD AA 3E      CALL CHKCMD   ;ANY PREVIOUS COMMAND?
5235      3DC3      C0 . .      RNZ          ;YES - ABORT ESCAPE SEQUENCE
5236      3DC4      70 . .      MOV M,B     ;NO - SET COMMAND CODE NUMBER
5237      3DC5      3A DE FF      LDA IODATA  ;FETCH LSB OF CURRENT PARAM
5238      3DC8      FE 0B .      CPI 11     ;PARAM TOO BIG?
5239      3DCA      F0 . .      RP        ;YES - ABORT ESC SEQ
5240      3DCB      32 D8 FF      STA IOCTYP  ;SAVE IT FOR AN INDEX
5241      3DCE      3A DF FF      LDA IODATA+1
5242      3DD1      B7 . .      ORA A      ;IS MSB ZERO?
5243      3DD2      C0 . .      RNZ          ;NO - ABORT ESCAPE SEQUENCE
5244      3DD3      C3 34 3E      JMP IOCEX0  ;YES - CLEAR PARM AND EXIT
5245      3DD6      . . .      ;
5246      3DD6      . . .      ; <W> - STORE A RECORD FROM THE DATACOM
5247      3DD6      . . .      ;
5248      3DD6      . . .      IOC120 EQU $
5249      3DD6      CD AA 3E      CALL CHKCMD   ;ANY PREVIOUS COMMAND?
5250      3DD9      C0 . .      RNZ          ;YES - ABORT ESCAPE SEQUENCE
5251      3DDA      36 05 .      MVI M,IOWNUM ;NO - SET COMMAND CODE NUMBER
5252      3DDC      2A DE FF      LHLD IODATA  ;TRANSFER PARAMETER TO
5253      3DDF      22 D5 FF      SHLD IOCCNT  ;COUNT FIELD
5254      3DE2      3A DD FF      LDA IOCSGN   ;TRANSFER SIGN
5255      3DE5      21 DC FF      LXI H,IOPSGN
5256      3DE8      77 . .      MOV M,A
5257      3DE9      B7 . .      ORA A      ;ANY PARAMETER RECEIVED?
5258      3DEA      C2 34 3E      JNZ IOCEX0  ;YES - CLEAR PARAM AND EXIT
5259      3DED      36 80 .      MVI M,200Q  ;NO - SET TO ABSOLUTE 0
5260      3DEF      C3 34 3E      JMP IOCEX0  ;CLEAR PARAM AND EXIT
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS
5262	3DF2	.	.	.	;
5263	3DF2	.	.	.	; <P> - CONTROL PARAMETER
5264	3DF2	.	.	.	;
5265	3DF2	.	.	.	IOC090 EQU \$
5266	3DF2	2A	DE	FF	LHLD IODATA ;TRANSFER PARAMETER TO COUNT
5267	3DF5	22	D5	FF	SHLD IOCCNT ;FIELD
5268	3DF8	3A	DD	FF	LDA IOCSGN ;TRANSFER SIGN
5269	3DF8	32	DC	FF	STA IOPSGN
5270	3DFE	C3	34	3E	JMP IOCEX0 ;CLEAR PARMETER AND EXIT
5271	3E01	.	.	.	;
5272	3E01	.	.	.	; <U> - DEVICE SPECIFICATION
5273	3E01	.	.	.	;
5274	3E01	.	.	.	IOC040 EQU \$
5275	3E01	CD	B0	3E	CALL DFNDEV ;IS IT A VALID DEVICE CODE?
5276	3E04	C0	.	.	RNZ ;NO - ABORT ESCAPE SEQUENCE
5277	3E05	32	DB	FF	STA IOCDEV ;PUT INTO DEVICE WORD
5278	3E08	C3	34	3E	JMP IOCEX0 ;CLEAR PARM AND EXIT
5279	3E08	.	.	.	;
5280	3E08	.	.	.	; <D> - DEFINE DESTINATION DEVICE
5281	3E08	.	.	.	;
5282	3E08	.	.	.	IOC080 EQU \$
5283	3E08	04	.	.	INR B ;INDEX TO IOCOUT
5284	3E0C	.	.	.	;
5285	3E0C	.	.	.	; <S> - DEFINE SOURCE DEVICE
5286	3E0C	.	.	.	;
5287	3E0C	.	.	.	IOC060 EQU \$
5288	3E0C	3A	F4	FF	LDA MDFLG1 ;IN EDIT MODE?
5289	3E0F	E6	10	.	ANI EDIT
5290	3E11	C0	.	.	RNZ ;YES - ABORT ESCAPE SEQUENCE
5291	3E12	78	.	.	MOV A,B
5292	3E13	C6	D8	.	ADI IOCINP-1 ;ADD BASE VALUE
5293	3E15	4F	.	.	MOV C,A
5294	3E16	CD	B0	3E	CALL DFNDEV ;IS IT A VALID DEVICE CODE?
5295	3E19	C0	.	.	RNZ ;NO - ABORT ESCAPE SEQUENCE
5296	3E1A	69	.	.	MOV L,C ;YES - MERGE WITH EXISTING
5297	3E1B	26	FF	.	MVI H,BASEH ;GET IOCINP/OUT HIGH ADDR
5298	3E1D	B6	.	.	ORA M ;DEVICE WORD
5299	3E1E	77	.	.	MOV M,A
5300	3E1F	C3	34	3E	JMP IOCEX0 ;CLEAR PARM AND EXIT

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 156
=====
5302      3E22      . . .      ;
5303      3E22      . . .      ; <^> - STATUS REQUEST
5304      3E22      . . .      ;
5305      3E22      . . .      IOC110 EQU $
5306      3E22      CD AA 3E      CALL CHKCMD ;ANY PREVIOUS COMMAND?
5307      3E25      C0 . .      RNZ ;YES - ABORT ESCAPE SEQUENCE
5308      3E26      36 07 .      MVI M,IOSNUM ;SET COMMAND CODE NUMBER
5309      3E28      CD 80 3E      CALL DFNDEV ;VALID DEVICE SPECIFIED?
5310      3E28      C2 32 3E      JNZ IOC115 ;NO - CHECK FOR NO SPEC
5311      3E2E      32 DB FF      STA IOCDEV ;YES - STORE DEVICE CODE
5312      3E31      AF . .      XRA A
5313      3E32      . . .      ;
5314      3E32      B7 . .      IOC115 ORA A ;ANY DEVICE SPECIFIED?
5315      3E33      C0 . .      RNZ ;YES - ERROR, ABORT SEQUENCE
5316      3E34      . . .      ; NO - CLEAR PARM AND EXIT
5317      3E34      . . .      ;
5318      3E34      . . .      ; CLEAR PARAMETER AND EXIT
5319      3E34      . . .      ;
5320      3E34      . . .      IOCEX0 EQU $
5321      3E34      21 00 00      LXI H,0 ;CLEAR THE PARAMETER
5322      3E37      22 DE FF      SHLD IODATA
5323      3E3A      97 . .      SUB A ;CLEAR THE SIGN
5324      3E3B      32 DD FF      STA IOCSGN
5325      3E3E      3A 88 FF      LDA CHAR ;FETCH THE CURRENT CHARACTER
5326      3E41      E6 20 .      ANI 40Q ;IS IT UPPER CASE?
5327      3E43      CA 4C 3E      JZ IOC130 ;YES - PROCESS COMMAND
5328      3E46      3E 02 .      MVI A,2 ;NO - RETURN TO ESC PROCCSO
5329      3E48      32 D1 FF      STA ESCFLG
5330      3E48      C9 . .      RET
5331      3E4C      . . .      ;
5332      3E4C      . . .      ; UPPER CASE CHARACTER FOUND - EXECUTE COMMAND
5333      3E4C      . . .      ;
5334      3E4C      . . .      IOC130 EQU $
5335      3E4C      21 DC FF      LXI H,IOPSGN ;ANY PARAMETER REC'D?
5336      3E4F      B6 . .      ORA M
5337      3E50      C2 58 3E      JNZ IOC140 ;YES -
5338      3E53      36 01 .      MVI M,1 ;NO - SET TO DEFAULT OF +1
5339      3E55      2E D5 .      MVI L,IOCCNT
5340      3E57      34 . .      INR M
5341      3E58      . . .      IOC140 EQU $
5342      3E58      CD 64 37      CALL SETDEV ;VALID DEVICE ASSIGNMENT?
5343      3E5B      D8 . .      RC ;NO - TERMINATE ESCAPE SEQ
5344      3E5C      CD 4F 00      CALL ESCEND ;END ESCAPE SEQUENCE
5345      3E5F      3A D7 FF      LDA IOCMND ;IS COMMAND A STATUS REQUEST
5346      3E62      FE 07 .      CPI IOSNUM
5347      3E64      3E F8 .      MVI A,-1-FILRED-USREAD-RDWOVT
5348      3E66      C4 2A 2B      CNZ CLIOFS ;IF NOT, CLEAR READ FLAGS
5349      3E69      C4 75 43      CNZ RDABR1 ;IF ANY CLEARED, ABORT READ
5350      3E6C      3A 6E FF      LDA DFLGS ;ESC SEQ FROM DATACOM?
5351      3E6F      E6 01 .      ANI SDACOM
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 157
5352	3E71	C2	85	3E	JNZ IOC150
5353	3E74	.	.	.	;*****
5354	3E74	.	.	.	; PERFORM LOCAL ESCAPE SEQUENCE *
5355	3E74	.	.	.	;*****
5356	3E74	3A	D7	FF	LDA IOCMND ;GET COMMAND CODE
5357	3E77	B7	.	.	ORA A ;ANY COMMAND EXCEPT DEV SEL?
5358	3E78	C8	.	.	RZ ;NO - RETURN IMMEDIATELY
5359	3E79	FE	05	.	CPI IOWNUM
5360	3E7B	D0	.	.	RNC ;IGNORE READ, WRITE, STATUS
5361	3E7C	21	9B	3D	LXI H,IOCMTB ;GET TABLE BASE ADDRESS
5362	3E7F	CD	9A	41	CALL INDJMP ;PERFORM FUNCTION
5363	3E82	C3	84	36	JMP USREXT ;REPORT ANY ERROR AND QUIT
5364	3E85	.	.	.	;*****
5365	3E85	.	.	.	; PERFORM REMOTE ESCAPE SEQUENCE *
5366	3E85	.	.	.	;*****
5367	3E85	.	.	.	IOC150 EQU \$
5368	3E85	01	FC	7E	LXI B,-1-SDVDUN-SDC2-SDVREC-SBINRY
5369	3E88	CD	55	00	CALL CLBLXF ;CLEAR OUTPUT PENDING FLAGS
5370	3E8B	21	9B	3D	LXI H,IOCMTB ;GET TABLE BASE ADDRESS
5371	3E8E	3A	D7	FF	LDA IOCMND ;GET COMMAND CODE
5372	3E91	B7	.	.	ORA A ;ANY COMMAND EXCEPT DEV SEL?
5373	3E92	C8	.	.	RZ ;NO - RETURN IMMEDIATELY
5374	3E93	FE	06	.	CPI IORNUM
5375	3E95	D2	9A	41	JNC INDJMP ;READ, STATUS DO NOT RET COD
5376	3E98	CD	9A	41	CALL INDJMP
5377	3E9B	DC	17	3D	CC FREBFS ;FREE BUFFERS ON ERROR
5378	3E9E	3A	4F	FF	LDA IOCERR ;SAVE COMPLETION CODE
5379	3EA1	32	4C	FF	STA IOCDPT
5380	3EA4	01	00	80	LXI B,SDVDUN ;SET UP TO TRANSFER
5381	3EA7	C3	5B	00	JMP SBLXFA

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 158
=====
5383      3EAA      . . .      ;
5384      3EAA      . . .      ; * * * * *
5385      3EAA      . . .      ;
5386      3EAA      . . .      ;   CHKCMD - CHECK FOR PRIOR COMMAND
5387      3EAA      . . .      ;
5388      3EAA      . . .      ;       EXIT :  Z = T, NO PRIOR COMMAND
5389      3EAA      . . .      ;                = F, COMMAND ALREADY GIVEN
5390      3EAA      . . .      ;                A = PRIOR COMMAND OR ZERO
5391      3EAA      . . .      ;
5392      3EAA      . . .      ;   CHKCMD EQU $
5393      3EAA      21  D7  FF      LXI  H,IOCMND  ;FETCH THE COMMAND WORD
5394      3EAD      7E . .      MOV  A,M
5395      3EAE      B7 . .      ORA  A          ;SET Z-FLAG
5396      3EAF      C9 . .      RET            ;RETURN
5397      3EB0      . . .      ;
5398      3EB0      . . .      ; * * * * *
5399      3EB0      . . .      ;
5400      3EB0      . . .      ;   DFNDEV - DEFINE DEVICE FROM PARAMETER VALUE
5401      3EB0      . . .      ;
5402      3EB0      . . .      ;       ENTRY:  H = BASEH
5403      3EB0      . . .      ;
5404      3EB0      . . .      ;       EXIT :  Z = F, INVALID DEVICE CODE
5405      3EB0      . . .      ;                A = 0, NO CODE SPECIFIED
5406      3EB0      . . .      ;                Z = T, DEVICE CODE VALID
5407      3EB0      . . .      ;                A = DEVICE BIT SETTING
5408      3EB0      . . .      ;                B,L DESTROYED
5409      3EB0      . . .      ;
5410      3EB0      . . .      ;   DFNDEV EQU $
5411      3EB0      2A  DE  FF      LHLD IODATA   ;FETCH PARAMETER VALUE
5412      3EB3      . . .      ;   DFNDEV EQU $
5413      3EB3      7D . .      MOV  A,L
5414      3EB4      B7 . .      ORA  A          ;WAS IT ZERO?
5415      3EB5      CA  C1  3E      JZ  DFN020    ;YES - CLEAR Z-FLAG AND EXIT
5416      3EB8      84 . .      ADD  H          ;ADD IN PARAMETER MSB
5417      3EB9      8D . .      CMP  L          ;IS MSB ZERO?
5418      3EBA      C0 . .      RNZ          ;NO - ERROR RETURN
5419      3EBB      47 . .      MOV  B,A       ;(PUT IN B FOR FNDB2)
5420      3EBC      D6  06 . .      SUI  6         ;IS VALUE IN RANGE?
5421      3EBE      DA  73  00      JC  FNDB2     ;YES - SET DEVICE BIT
5422      3EC1      . . .      ;   DFN020 EQU $
5423      3EC1      FE  01 . .      CPI  1         ;SET NZ (1 IS A VALID CODE)
5424      3EC3      C9 . .      RET            ;RETURN
=====

```

```

=====
ITEM   LOC   OBJECT CODE  SOURCE STATEMENTS                                     PAGE 159
=====
5426   3EC4   . . . ;
5427   3EC4   . . . ; * * * * *
5428   3EC4   . . . ;
5429   3EC4   . . . ; XFRREC - TRANSFER ONE RECORD
5430   3EC4   . . . ;
5431   3EC4   . . . ; ENTRY: H = BASE
5432   3EC4   . . . ;
5433   3EC4   . . . ; EXITS TO SYSTEM VIA "IODONE" OR "IOFAIL"
5434   3EC4   . . . ;
5435   3EC4   . . . XFRREC EQU $
5436   3EC4   05 . . . DCR B
5437   3EC5   . . . ;
5438   3EC5   . . . ; * * * * *
5439   3EC5   . . . ;
5440   3EC5   . . . ; XFREOF - TRANSFER TO END OF FILE MARK
5441   3EC5   . . . ;
5442   3EC5   . . . XFREOF EQU $
5443   3EC5   05 . . . DCR B
5444   3EC6   . . . ;
5445   3EC6   . . . ; * * * * *
5446   3EC6   . . . ;
5447   3EC6   . . . ; XFREVD - TRANSFER TO END OF VALID DATA MARK
5448   3EC6   . . . ;
5449   3EC6   . . . XFREVD EQU $
5450   3EC6   3A DB FF LDA IOCTYP ;GET COMMAND MODIFIER
5451   3EC9   0F . . . RRC ;SET FOR COMPARE OPERATION?
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 160
5453	3ECA	.	.	;	
5454	3ECA	.	.	; * * * * *	
5455	3ECA	.	.	;	
5456	3ECA	.	.	; XFRD2D - TRANSFER FROM DEVICE TO DEVICE	
5457	3ECA	.	.	; OR COMPARE BETWEEN DEVICES	
5458	3ECA	.	.	;	
5459	3ECA	.	.	; ENTRY : C => COMPARE	
5460	3ECA	.	.	; NC => TRANSFER	
5461	3ECA	.	.	; XFRD2D : B = TRANSFER LIMIT	
5462	3ECA	.	.	; XFR001 : A = TRANSFER LIMIT	
5463	3ECA	.	.	; H,L = INPUT, OUTPUT DEVICES	
5464	3ECA	.	.	;	
5465	3ECA	.	.	; EXIT : NC => NO ERROR	
5466	3ECA	.	.	; C => ERROR	
5467	3ECA	.	.	;	
5468	3ECA	.	.	;	
5469	3ECA	.	.	XFRD2D EQU \$	
5470	3ECA	2A	4D	FF LHL D OUTDEV ;GET NORMAL INPUT,OUTPUT DEV	
5471	3ECD	78	.	MOV A,B	
5472	3ECE	.	.	XFR001 EQU \$	
5473	3ECE	11	76	3F LXI D,XFRFCN	
5474	3ED1	32	46	FF STA CMLIM	
5475	3ED4	D2	F4	3E JNC XFR100 ;IF NOT COMPARE, SKIP	
5476	3ED7	.	.	;	
5477	3ED7	.	.	;*****	
5478	3ED7	.	.	; SET UP FOR COMPARE OPERATION *	
5479	3ED7	EB	.	;	
5480	3ED8	21	94	3B LXI H,TMTMSG ;"TO MANY TO DEVICES"	
5481	3ED8	97	.	SUB A ;MORE THAN ONE OUTPUT DEV?	
5482	3EDC	93	.	SUB E	
5483	3EDD	A3	.	ANA E	
5484	3EDE	BB	.	CMP E	
5485	3EDF	C2	80	41 JNZ IOFAI1 ;YES - REPORT ERROR	
5486	3EE2	EB	.	XCHG	
5487	3EE3	7D	.	MOV A,L ;IS OUTPUT DISPLAY?	
5488	3EE4	FE	04	CPI DISPLY	
5489	3EE6	C2	EC	3E JNZ XFR050 ;NO -	
5490	3EE9	7C	.	MOV A,H ;YES - SWAP INPUT & OUTPUT	
5491	3EEA	65	.	MOV H,L	
5492	3EEB	6F	.	MOV L,A	
5493	3EEC	.	.	XFR050 EQU \$	
5494	3EEC	3E	FF	MVI A,-1 ;CMLIM <- -1 => RECORD	
5495	3EEE	01	00	01 LXI B,400Q ;B = FILE 1; C = RECORD 0	
5496	3EF1	11	7C	3F LXI D,CMPFCN	
5497	3EF4	.	.	;	
5498	3EF4	.	.	; *****	
5499	3EF4	.	.	; FINISHED SETTING UP COMPARE *	
5500	3EF4	.	.	;	
5501	3EF4	32	47	FF XFR100 EQU \$	
5502	3EF7	22	22	FF STA XFRLLIM	
				SHLD SAVOUT ;SAVE INPUT,OUTPUT DEVICES	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 161
5503	3EFA	7C	.	MOV A,H ;INPUT = OUTPUT?	
5504	3EFB	A5	.	ANA L	
5505	3EFC	C2	AD 41	JNZ IOFAIO ;YES - REORT ERROR	
5506	3EFF	7C	.	MOV A,H ;ANY TAPES INVOLVED?	
5507	3F00	B5	.	ORA L	
5508	3F01	E6	03 .	ANI LFTCTU+RGCTU	
5509	3F03	C4	C7 2D	CNZ BSYCHK ;YES - WAIT TILL NOT BUSY	
5510	3F06	D8	.	RC ;RETURN ON USER INTERRUPT	
5511	3F07	D5	.	PUSH D ;SAVE FUNCTION ADDRESS	
5512	3F08	.	.	*****	
5513	3F08	.	.	; DECIDE WHETHER TO INHIBIT ROLLUP *	
5514	3F08	.	.	*****	
5515	3F08	.	.	; INHIBIT ROLLUP IFF...	
5516	3F08	3A	23 FF	LDA SAVINP ;DISPLAY IS INPUT DEVICE	
5517	3F0B	FE	04 .	CPI DISPLY	
5518	3F0D	C2	26 3F	JNZ XFR200	
5519	3F10	C5	.	PUSH B	
5520	3F11	3E	EF .	MVI A,-1-RECINI	
5521	3F13	CD	2A 2B	CALL CLIOFS ;(INITIALIZE DISPLAY IFF	
5522	3F16	CC	A9 44	CZ INTDSP ;NOT DOING RECORD)	
5523	3F19	C1	.	POP B	
5524	3F1A	3A	46 FF	LDA CMLIM ;DOING FILE COPY,	
5525	3F1D	B7	.	ORA A	
5526	3F1E	CC	CC 47	CZ CHKMT ;NON-FORMAT MODE	
5527	3F21	3E	20 .	MVI A,RECPGE ;INHIBIT ROLLUP	
5528	3F23	CC	24 2B	CZ STIOFS	
5529	3F26	.	.	*****	
5530	3F26	.	.	; PROCESS ONE RECORD *	
5531	3F26	.	.	*****	
5532	3F26	.	.	XFR200 EQU \$;GET A RECORD FROM INPUT	
5533	3F26	C5	.	PUSH B ;SAVE FILE AND REC COUNTER	
5534	3F27	3A	23 FF	LDA SAVINP ;GET FIRST DEVICE	
5535	3F2A	CD	67 41	CALL GETIO1 ;GET A REC FROM THE DEVICE	
5536	3F2D	C1	.	POP B	
5537	3F2E	DA	4C 3F	JC XFR500 ;EXIT ON ERROR	
5538	3F31	1B	.	DCX D ;D,E -> TYPE	
5539	3F32	1A	.	LDAX D ;GET TYPE	
5540	3F33	13	.	INX D	
5541	3F34	FE	02 .	CPI 2 ;DISPLAY BOUNDARY?	
5542	3F36	F2	5C 3F	JP XFR600 ;YES - GO CHECK IT OUT	
5543	3F39	.	.	XFR220 EQU \$;OUTPUT OR CMP THE RECORD	
5544	3F39	E1	.	POP H ;H,L -> FUNCTION	
5545	3F3A	E5	.	PUSH H	
5546	3F3B	CF	.	RST RSTJMP	
5547	3F3C	DA	4C 3F	JC XFR500 ;EXIT ON ERROR	
5548	3F3F	1B	.	DCX D ;D,E -> TYPE	
5549	3F40	1A	.	LDAX D ;RECALL RECORD TYPE	
5550	3F41	13	.	INX D	
5551	3F42	.	.	XFR250 EQU \$	
5552	3F42	21	46 FF	LXI H,CMLIM ;COMP REC TYPE TO XFR LIMIT	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 162
5553	3F45	96	.	.	SUB M ;LIMIT REACHED?	
5554	3F46	FA	26	3F	JM XFR200 ;NO - DO NEXT RECORD	
5555	3F49	E1	.	.	POP H	
5556	3F4A	B7	.	.	ORA A ;YES - RETURN SUCCESS	
5557	3F48	C9	.	.	RET	
5558	3F4C	.	.	.	XFR500 EQU \$;ERROR EXIT	
5559	3F4C	3E	DF	.	MVI A,-1-RECPGE ;CLEAR INHIBIT ROLLUP FLA	
5560	3F4E	CD	2A	2B	CALL CLIOFS	
5561	3F51	3A	23	FF	LDA SAVINP ;WAS INPUT DEVICE A TAPE?	
5562	3F54	E6	03	.	ANI LFTCTU+RGCTU	
5563	3F56	C4	E4	2B	CNZ STOPTP ;IF SO, STOP IT	
5564	3F59	E1	.	.	POP H	
5565	3F5A	37	.	.	STC ;RETURN C => ERROR	
5566	3F5B	C9	.	.	RET	
5567	3F5C	.	.	.	;	
5568	3F5C	.	.	.	;	
5569	3F5C	.	.	.	;	
5570	3F5C	.	.	.	;	
5571	3F5C	.	.	.	;	
5572	3F5C	.	.	.	;	
5573	3F5C	.	.	.	;	
5574	3F5C	.	.	.	;	
5575	3F5C	CD	CC	47	XFR600 EQU \$	
5576	3F5F	C2	6C	3F	CALL CHKFMT ;IN FORMAT MODE?	
5577	3F62	97	.	.	JNZ XFR650 ;YES - OUTPUT END OF FILE	
5578	3F63	12	.	.	SUB A ;NO - RELEASE BUFFER	
5579	3F64	1B	.	.	STAX D	
5580	3F65	1A	.	.	DCX D ;GET RECORD TYPE	
5581	3F66	13	.	.	LDAX D	
5582	3F67	D6	02	.	INX D	
5583	3F69	C3	42	3F	SUI 2 ;CONVERT TO NORMAL TYPE	
5584	3F6C	.	.	.	JMP XFR250 ;COMPARE WITH XFR LIMIT	
5585	3F6C	1B	.	.	XFR650 EQU \$	
5586	3F6D	97	.	.	DCX D ;D,E -> TYPE	
5587	3F6E	12	.	.	SUB A ;SET TYPE = EOF	
5588	3F6F	13	.	.	STAX D	
5589	3F70	32	46	FF	INX D	
5590	3F73	C3	39	3F	STA CMPLIM ;SET TRANSFER LIMIT TO EOF	
5591	3F76	.	.	.	JMP XFR220 ;OUTPUT THE BUFFER	
5592	3F76	.	.	.	;	
5593	3F76	.	.	.	;	
5594	3F76	.	.	.	;	
5595	3F76	.	.	.	;	
5596	3F76	3A	22	FF	XFRFCN EQU \$	
5597	3F79	C3	7D	41	LDA SAVOUT ;GET OUTPUT DEVICE(S)	
5598	3F7C	.	.	.	JMP PUTIO1 ;OUTPUT THE RECORD	
5599	3F7C	.	.	.	;	
5600	3F7C	.	.	.	;	
5601	3F7C	.	.	.	;	
5602	3F7C	.	.	.	;	
					CMPFCN EQU \$	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS
5603	3F7C	0C	.	.	INR C ;INCREMENT RECORD COUNT
5604	3F7D	B7	.	.	ORA A ;FILE MARK RECORD?
5605	3F7E	FA	84	3F	JM CMP100 ;NO - DO NEXT RECORD
5606	3F81	04	.	.	INR B ;YES - INCREMENT FILE COUNT
5607	3F82	0E	00	.	MVI C,0 ;AND CLEAR RECORD COUNT
5608	3F84	.	.	.	CMP100 EQU \$
5609	3F84	C5	.	.	PUSH B
5610	3F85	3A	22	FF	LDA SAVOUT ;GET SECOND DEVICE
5611	3F88	CD	67	41	CALL GETIO1 ;GET A RECORD FROM THE DEVIC
5612	3F8B	C1	.	.	POP B
5613	3F8C	D8	.	.	RC ;REPORT INPUT ERRORS
5614	3F8D	D5	.	.	PUSH D ;SAVE A STATUS POINTER
5615	3F8E	CD	C3	32	CALL CMPBFS ;COMPARE THE RECORDS
5616	3F91	D1	.	.	POP D
5617	3F92	DA	B3	41	JC IOFAIL ;ON DIFFERENCE, SET ERROR FL
5618	3F95	C3	17	3D	JMP FREBFS ;SUCCESS - FREE BUFFERS

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 164
=====
5620      3F98      .      .      .      ;
5621      3F98      .      .      .      ; * * * * *
5622      3F98      .      .      .      ;
5623      3F98      .      .      .      ; CTRLIO - PERFORM I/O CONTROL FUNCTION
5624      3F98      .      .      .      ;
5625      3F98      .      .      .      ;
5626      3F98      .      .      .      ; EXITS TO SYSTEM VIA "ESCEND"
5627      3F98      .      .      .      ;
5628      3F98      .      .      .      CTRLIO EQU $
5629      3F98      21  DB  FF      LXI H,IOCDEV ;FETCH DEVICE PARAMETER
5630      3F98      7E      .      .      MOV A,M
5631      3F9C      B7      .      .      ORA A ;DEVICE SPECIFIED?
5632      3F9D      C2  AF  3F      JNZ CTR020 ;YES - EXECUTE FUNCTION
5633      3FA0      2E  D8  .      MVI L,IOCTYP ;NO - FETCH COMMAND CODE
5634      3FA2      7E      .      .      MOV A,M
5635      3FA3      06  05  .      SUI 5 ;DOES COMMAND DEFAULT TO
5636      3FA5      2E  4E  .      MVI L,INPDEV ;SOURCE DEVICE?
5637      3FA7      DA  AB  3F      JC CTR010 ;YES - EXECUTE FUNCTION
5638      3FAA      2D      .      .      DCR L ;NO - USE OUTPUT DEVICES
5639      3FAB      .      .      .      CTR010 EQU $
5640      3FAB      7E      .      .      MOV A,M ;FETCH DEVICE FLAG SETTINGS
5641      3FAC      2E  DB  .      MVI L,IOCDEV ;SET FOR OBJECT DEVICE
5642      3FAE      77      .      .      MOV M,A
5643      3FAF      .      .      .      ;
5644      3FAF      .      .      .      ; EXECUTE DEVICE CONTROL
5645      3FAF      .      .      .      ;
5646      3FAF      .      .      .      CTR020 EQU $
5647      3FAF      2E  DC  .      MVI L,IOPSGN
5648      3FB1      7E      .      .      MOV A,M ;FETCH THE SIGN
5649      3FB2      B7      .      .      ORA A ;VALUE SPECIFIED?
5650      3FB3      C2  BE  3F      JNZ CTR025 ;YES - EXECUTE FUNCTION
5651      3FB6      36  01  .      MVI M,1 ;NO - SET DEFAULT TO +1
5652      3FB8      21  01  00      LXI H,1
5653      3FB8      22  D5  FF      SHLD IOCCNT
5654      3FBE      .      .      .      CTR025 EQU $
5655      3FBE      21  4C  FF      LXI H,IOC DPT ;INITIALIZE DEVICE FLAG
5656      3FC1      3E  01  .      MVI A,1
5657      3FC3      .      .      .      ;
5658      3FC3      .      .      .      CTR030 EQU $
5659      3FC3      77      .      .      MOV M,A ;STORE DEVICE FLAG
5660      3FC4      2E  DB  .      MVI L,IOCDEV ;COMPARE TO SELECTED DEVICES
5661      3FC6      A6      .      .      ANA M ;DEVICE SELECTED?
5662      3FC7      CA  D1  3F      JZ CTR040 ;NO - TRY ANOTHER DEVICE
5663      3FCA      21  4D  41      LXI H,CTLTAB ;SET TABLE BASE ADDRESS
5664      3FCD      CD  94  41      CALL SETJMP ;PERFORM INDIRECT CALL
5665      3FD0      D8      .      .      RC ;RETURN ON ERROR
5666      3FD1      .      .      .      ; NO - TRY NEXT DEVICE
5667      3FD1      .      .      .      CTR040 EQU $
5668      3FD1      21  4C  FF      LXI H,IOC DPT ;RECALL DEVICE POINTER
5669      3FD4      7E      .      .      MOV A,M

```



```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 165
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 165
5670	3FD5	07 . .	RLC	;ALL DEVICES SCANNED?
5671	3FD6	3F . .	CMC	
5672	3FD7	D0 . .	RNC	;YES - RETURN SUCCESS
5673	3FD8	C3 C3 3F	JMP CTR030	;NO - DO NEXT

```
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 166
5675	3FDB	.	.	.	;
5676	3FDB	.	.	.	; * * * * *
5677	3FDB	.	.	.	;
5678	3FDB	.	.	.	;
5679	3FDB	.	.	.	TPSTAT - DISPLAY CURRENT FILE AND INCHES
5680	3FDB	.	.	.	LEFT FOR EACH TAPE
5681	3FDB	.	.	.	;
5682	3FDB	.	.	.	ENTRY FROM IOCKEY (GREEN KEY)
5683	3FDB	.	.	.	;
5684	3FDB	.	.	.	EXIT TO MAIN CODE
5685	3FDB	.	.	.	;
5686	3FDB	.	.	.	DESTROYS ALL REGISTERS
5687	3FDB	.	.	.	TPSTAT EQU \$
5688	3FDB	CD	C7	2D	CALL BSYCHK ;WAIT UNTIL TAPES FREE
5689	3FDE	D8	.	.	RC ;RETURN ON USER INTERRUPT
5690	3FDF	21	BE	3B	LXI H,NLTPMS
5691	3FE2	22	ED	FF	SHLD MSGPT3
5692	3FES	21	93	3C	LXI H,CUMAMS
5693	3FE8	22	EB	FF	SHLD MSGPT4
5694	3FEB	21	CE	3B	LXI H,NRTPMS
5695	3FEE	22	E7	FF	SHLD MSGPT6
5696	3FF1	21	BD	3B	LXI H,EOPMSG
5697	3FF4	22	E5	FF	SHLD MSGPT7
5698	3FF7	CD	8D	2D	CALL SELLECT
5699	3FFA	.	.	.	;*****
5700	3FFA	.	.	.	; ROM BREAK 3
5701	3FFA	C3	02	40	JMP ZBRK3C
5702	3FFD	.	.	.	ORG ZBRK2+40000
5703	4000	.	.	.	ZBRK3 EQU \$
5704	4000	54	.	.	DB VERSN ;ROM PRESENT/VERSION FLAG
5705	4001	40	.	.	DB ZBRK3/256
5706	4002	.	.	.	ZBRK3C EQU \$
5707	4002	.	.	.	;*****
5708	4002	CD	FF	3C	CALL GTIOB0 ;GET A BUFFER FOR COUNTS
5709	4005	D8	.	.	RC ;RETURN ON USER INTERRUPT
5710	4006	CD	2A	3D	CALL GETPTR ;GET POINTER TO THE BUFFER
5711	4009	22	EF	FF	SHLD MSGPT2
5712	400C	CD	20	40	CALL TPS100 ;GET ASCII FOR LEFT COUNTS
5713	400F	EB	.	.	XCHG ;SAVE BUFFER POINTER WHILE
5714	4010	CD	BA	2D	CALL SELRCT ;SWAPPING UNITS
5715	4013	EB	.	.	XCHG
5716	4014	22	E9	FF	SHLD MSGPT5
5717	4017	CD	20	40	CALL TPS100 ;GET ASCII FOR RIGHT COUNT
5718	401A	21	42	3B	LXI H,FLINMS ;GET FIRST PART OF MESSAGE
5719	401D	C3	B0	41	JMP IOFAI1 ;MAKE SURE MESSAGE IS DISPLY
5720	4020	.	.	.	;*****
5721	4020	.	.	.	; IF TAPE IS INSERTED, DISPLAY FILE AND INCHES *
5722	4020	.	.	.	;*****
5723	4020	.	.	.	TPS100 EQU \$
5724	4020	CD	DC	2A	CALL GTCTBT ;GET BIT FOR SELECTED UNIT

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 167
5725	4023	47	.	.	MOV B,A	
5726	4024	3A	66	FF	LDA CTSTAT ;GET CURRENT STATUS	
5727	4027	A0	.	.	ANA B ;TAPE INSERTED?	
5728	4028	C2	34	40	JNZ TPS120 ;YES -	
5729	4028	32	5E	FF	STA FILNUM ;NO - CLEAR FILE NUMBER	
5730	402E	32	5F	FF	STA ABSTAK ;AND TACH COUNTER	
5731	4031	32	60	FF	STA ABSTAK+1	
5732	4034	.	.	.	*****	
5733	4034	.	.	.	; CHANGE CURRENT FILE TO ASCII *	
5734	4034	.	.	.	*****	
5735	4034	.	.	.	TPS120 EQU \$	
5736	4034	3A	5E	FF	LDA FILNUM ;GET CURRENT FILE NUMBER	
5737	4037	CD	AB	00	CALL BN2DE0 ;CONVERT TO ASCII	
5738	403A	36	80	.	MVI M,200Q ;WRITE ' '	
5739	403C	23	.	.	INX H	
5740	403D	36	20	.	MVI M,40Q	
5741	403F	23	.	.	INX H	
5742	4040	36	8A	.	MVI M,212Q	
5743	4042	23	.	.	INX H	
5744	4043	.	.	.	*****	
5745	4043	.	.	.	; CHANGE TACH TO ASCII INCHES LEFT *	
5746	4043	.	.	.	*****	
5747	4043	E5	.	.	PUSH H ;SAVE BUFFER POINTER	
5748	4044	97	.	.	SUB A ;INSURE THAT COUNTER IS NOT	
5749	4045	CD	8E	2A	CALL OCM001 ;INVERTED	
5750	4048	.	.	.	TPS130 EQU \$	
5751	4048	2A	5F	FF	LHLD ABSTAK ;H,L = TACH COUNT	
5752	404B	3E	C3	.	MVI A,-STRTAK/256+3+2	
5753	404D	.	.	.	; 3 = BOT-LP DISTANCE, 2 = MARGIN FOR ERROR	
5754	404D	BC	.	.	CMP H ;HAS COUNT OVERFLOWED?	
5755	404E	D2	54	40	JNC TPS140 ;NO -	
5756	4051	21	00	00	LXI H,0 ;YES - DISPLAT "0"	
5757	4054	.	.	.	TPS140 EQU \$	
5758	4054	11	FF	FF	LXI D,-1 ;COUNT FOR RECORDS	
5759	4057	01	E3	FF	LXI B,-29 ;29 TACHS/INCH	
5760	405A	.	.	.	TPS160 EQU \$	
5761	405A	13	.	.	INX D ;INCREMENT COUNTER	
5762	405B	09	.	.	DAD B ;SUBTRACT TACHS FOR ONE REC	
5763	405C	DA	5A	40	JC TPS160 ;CONTINUE IF MORE TACHS	
5764	405F	E1	.	.	POP H ;RECALL LOC FOR ASCII	
5765	4060	CD	A8	00	CALL BN2DEC ;CONVERT COUNT TO ASCII	
5766	4063	36	00	.	MVI M,0 ;SIGNAL END OF MESSAGE	
5767	4065	23	.	.	INX H	
5768	4066	C9	.	.	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 168
5770	4067	.	.	.	;
5771	4067	.	.	.	; * * * * *
5772	4067	.	.	.	;
5773	4067	.	.	.	; IOREAD - READ ONE RECORD TO DATACOM
5774	4067	.	.	.	;
5775	4067	.	.	.	;
5776	4067	.	.	.	;
5777	4067	.	.	.	IOREAD EQU \$
5778	4067	01	01	00	LXI B,SDVREC ;GET FLAG TO BE SET
5779	406A	21	64	FF	LXI H,IOFLG2 ;SET EXTERNAL READ FLAG
5780	406D	36	01	.	MVI M,EXTB2D
5781	406F	2E	47	.	MVI L,XFRLIM ;SET DEFAULT LIMIT TO ONE
5782	4071	36	FF	.	MVI M,-1 ;RECORD
5783	4073	3A	D8	FF	LDA IOCTYP ;GET COMMAND MODIFIER
5784	4076	E6	06	.	ANI FILRED+BINXMT
5785	4078	FE	04	.	CPI FILRED ;FILE OR BINARY READ?
5786	407A	DA	58	00	JC SBLXF0 ;RECORD - SET PENDING FLAG
5787	407D	32	D8	FF	STA IOCTYP ;FILE - TURN OFF RE-TRANSMIT
5788	4080	36	00	.	MVI M,0 ;SET XFER LIMIT FOR 1 FILE
5789	4082	CD	B1	00	CALL GTMODE ;PAGE MODE?
5790	4085	CA	8A	40	JZ IRD010 ;NO - DON'T SET "RDWOWT"
5791	4088	3E	01	.	MVI A,RDWOWT ;YES - PERFORM FILE READ
5792	408A	.	.	.	IRD010 EQU \$;WITHOUT WAIT
5793	408A	F6	04	.	ORI FILRED ;SET FILE READ FLAG
5794	408C	CD	24	2B	CALL STIOFS
5795	408F	C3	58	00	JMP SBLXF0 ;TO "IORDGO" AFTER HANDSHK

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
5797	4092	.	.	.
5798	4092	.	.	.
5799	4092	.	.	.
5800	4092	.	.	IOSTAT - GET DEVICE STATUS
5801	4092	.	.	.
5802	4092	.	.	.
5803	4092	.	.	RETURNS TO SYSTEM VIA "ESCEND"
5804	4092	.	.	.
5805	4092	.	.	IOSTAT EQU \$
5806	4092	01	00	08 LXI B,SDVST ;SET DEVICE STATUS PENDING
5807	4095	CD	58	00 CALL SBLXFO ;FLAG
5808	4098	3A	DB	FF LDA IOCDEV ;FETCH DEVICE CODE
5809	409B	B7	.	. ORA A ;DEVICE SPECIFIED?
5810	409C	C2	A2	40 JNZ IOS010 ;YES - GET ITS STATUS
5811	409F	3A	4E	FF LDA INPDEV ;NO - DEFAULT TO SOURCE
5812	40A2	.	.	.
5813	40A2	32	48	FF STA IOSTA0 ;STORE DEVICE CODE
5814	40A5	21	57	41 LXI H,STATTB ;SET TABLE BASE ADDRESS
5815	40A8	C3	94	41 JMP SETJMP ;PERFORM INDIRECT JUMP

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 170
5817	40AB	.	.	.	;
5818	40AB	.	.	.	; * * * * *
5819	40AB	.	.	.	;
5820	40AB	.	.	.	; IOWRIT - WRITE DATACOM ONTO DEVICE
5821	40AB	.	.	.	;
5822	40AB	.	.	.	;
5823	40AB	.	.	.	; READS ONE RECORD FROM THE DATACOM AND
5824	40AB	.	.	.	; STORES IT ON ALL DESTINATION DEVICES
5825	40AB	.	.	.	;
5826	40AB	.	.	.	; DESTROYS ALL REGISTERS.
5827	40AB	.	.	.	;
5828	40AB	.	.	.	IOWRIT EQU \$
5829	40AB	CD	FF	3C	CALL GTIOB0 ;GET AN I/O BUFFER
5830	40AE	D8	.	.	RC ;RETURN ON ERROR
5831	40AF	36	20	.	MVI M,DATCOM ;MARK FOR INPUT FROM DATACOM
5832	40B1	2B	.	.	DCX H ;H,L -> TYPE
5833	40B2	36	FF	.	MVI M,-1 ;TYPE = DATA RECORD
5834	40B4	2B	.	.	DCX H ;H,L -> LENGTH
5835	40B5	EB	.	.	XCHG ;D,E -> LENGTH
5836	40B6	2A	D5	FF	LHLD IOCCNT ;GET PARAMETER
5837	40B9	4D	.	.	MOV C,L ;C IS USED AS COUNTER
5838	40BA	7D	.	.	MOV A,L ;ANY COUNT SPECIFIED?
5839	40BB	12	.	.	STAX D ;(SAVE COUNT IN BUFFER SIZE)
5840	40BC	B4	.	.	ORA H
5841	40BD	C2	C8	40	JNZ IOW020 ;YES - DO BINARY LOAD
5842	40C0	CD	EB	40	CALL DC2BUF ;NO - LOAD ASCII RECORD
5843	40C3	D8	.	.	RC ;RETURN ON ERROR
5844	40C4	EB	.	.	XCHG ;D,E -> LENGTH
5845	40C5	C3	E3	40	JMP IOW030 ;STORE THE RECORD
5846	40C8	.	.	.	;*****
5847	40C8	.	.	.	; BINARY TYPE LOAD - LOAD 8-BIT CHARACTERS *
5848	40C8	.	.	.	;*****
5849	40C8	.	.	.	IOW020 EQU \$
5850	40C8	CD	2A	3D	CALL GETPTR ;GET POINTER TO 1ST BUF BYTE
5851	40CB	.	.	.	IOW023 EQU \$
5852	40CB	E5	.	.	PUSH H ;SAVE REGS KILLED BY MONITOR
5853	40CC	CD	C1	29	CALL CTMON1 ;ANY CTU ERRORS?
5854	40CF	D4	2E	48	CNC RETSCN ;NO - RETURN KEY PRESSED?
5855	40D2	E1	.	.	POP H ;RESTORE REGISTERS
5856	40D3	D8	.	.	RC ;RETURN ON ERROR
5857	40D4	.	.	.	IOW025 EQU \$
5858	40D4	CD	1D	50	CALL ZGTBIN ;GET A BINARY BYTE
5859	40D7	DA	E2	47	JC DCERR1 ;QUIT ON DATACOMM ERROR
5860	40DA	C2	CB	40	JNZ IOW023 ;CHK CTU & TRY AGAIN ON WAIT
5861	40DD	77	.	.	MOV M,A ;STORE BYTE
5862	40DE	23	.	.	INX H ;INCREMENT BUFFER POINTER
5863	40DF	0D	.	.	DCR C ;RECORD DONE?
5864	40E0	C2	D4	40	JNZ IOW025 ;NO - GET NEXT CHAR
5865	40E3	.	.	.	;*****
5866	40E3	.	.	.	; STORE THE RECORD *

13255
2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 171
=====
5867      40E3      . . .      ;*****
5868      40E3      . . .      IOW030 EQU $
5869      40E3      13 . .      INX D      ;WANT D,E -> STATUS
5870      40E4      13 . .      INX D
5871      40E5      C3 7A 41     JMP PUTIO   ;OUTPUT TO ALL "OUT" DEVICES
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 172
=====
5873     40E8      . . .      ;
5874     40E8      . . .      ; * * * * *
5875     40E8      . . .      ;
5876     40E8      . . .      ;          DC2BUF = READ ASCII RECORD FROM DATACOM TO
5877     40E8      . . .      ;          I/O BUFFER
5878     40E8      . . .      ;
5879     40E8      . . .      ;          THIS ROUTINE LOADS AN I/O BUFFER WITH
5880     40E8      . . .      ;          NORMAL ASCII CHARACTERS FROM THE DATACOM
5881     40E8      . . .      ;          (GETDC). IT RETURNS WHEN A LF IS RECEIVED,
5882     40E8      . . .      ;          OR 256 BYTES HAVE BEEN RECEIVED, OR WHEN A
5883     40E8      . . .      ;          DATACOM ERROR IS DETECTED.
5884     40E8      . . .      ;
5885     40E8      . . .      ;          ENTRY:  BUFFER STATUS SET
5886     40E8      . . .      ;          BUFFER TYPE = NORMAL (DATA) RECORD
5887     40E8      . . .      ;          D,E -> BUFFER LENGTH (SET TO 0)
5888     40E8      . . .      ;          IOFLG2[RECINI] = 1 => 1ST CHAR FOR
5889     40E8      . . .      ;          BUFFER IS IN "CHAR" (RECORD-
5890     40E8      . . .      ;          MODE INITIALIZATION).
5891     40E8      . . .      ;
5892     40E8      . . .      ;          EXIT :  IOFLG2[RECINI] = 0
5893     40E8      . . .      ;          EXIT :  NC => NO ERROR
5894     40E8      . . .      ;          H,L -> BUFFER LENGTH
5895     40E8      . . .      ;          A,D,E DESTROYED
5896     40E8      . . .      ;          C => ERROR, OR USER INTERRUPT
5897     40E8      . . .      ;          A,D,E,H,L DESTROYED
5898     40E8      . . .      ;
5899     40E8      . . .      ;          DC2BUF EQU $
5900     40E8      3E EF .      MVI A,-1-RECINI ;TEST AND CLEAR RECINI
5901     40EA      CD 2A 2B      CALL CLIOFS      ;FIRST CHAR IN "CHAR"?
5902     40ED      3A 88 FF      LDA CHAR         ;GET CHAR IF SO
5903     40F0      F5 . .      PUSH PSW         ;SAVE FLAG AND CHAR
5904     40F1      CD 2A 3D      CALL GETPTR      ;GET POINTER TO BUFFER
5905     40F4      EB . .      XCHG            ;D,E -> BUF; H,L -> LENGTH
5906     40F5      F1 . .      POP PSW         ;RECALL FLAG AND CHAR
5907     40F6      C2 0B 41      JNZ DCB030      ;IF FIRST CHAR IN A, STORE I
5908     40F9      . . .      DCB010 EQU $
5909     40F9      E5 . .      PUSH H          ;SAVE REGS KILLED BY MONITOR
5910     40FA      CD C1 29      CALL CTMON1     ;CHECK FOR CTU ERRORS
5911     40FD      D4 31 48      CNC RETSC0     ;IF OK, MONITOR KYBD
5912     4100      E1 . .      POP H
5913     4101      D8 . .      RC              ;RET ON ERROR OR USER INTRUP
5914     4102      . . .      DCB020 EQU $
5915     4102      CD 17 50      CALL ZGETDC     ;CHECK DATACOM FOR CHAR
5916     4105      DA E2 47      JC DCERR1      ;PROCESS DATACOMM ERRORS
5917     4108      C2 F9 40      JNZ DCB010     ;WAIT - MUN CTU & KYBD, RETR
5918     410B      . . .      DCB030 EQU $
5919     410B      12 . .      STAX D
5920     410C      13 . .      INX D          ;INCREMENT BUF POINTER
5921     410D      34 . .      INR M          ;INCREMENT LENGTH COUNTER
5922     410E      C8 . .      RZ            ;RETURN IF LENGTH = 256
=====

```



```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 173
=====
5923     410F     FE 0A .      CPI LF
5924     4111     C8 . .      RZ ;RETURN IF LF
5925     4112     C3 02 41    JMP DCB020 ;GET NEXT CHARACTER
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 174
=====
5927      4115      . . .      ;*****
5928      4115      . . .      ; ALTERNATE I/O DEVICE DRIVERS *
5929      4115      . . .      ;*****
5930      4115      . . .      STALT EQU $ ;DEVICE STATUS
5931      4115      21 49 FF    LXI H,IUSTA1 ;CLEAR STATUS AREA
5932      4118      97 . .     SUB A ;IN CASE CODE IS
5933      4119      77 . .     MOV M,A ;NOT THERE
5934      411A      23 . .     INX H
5935      411B      77 . .     MOV M,A
5936      411C      23 . .     INX H
5937      411D      77 . .     MOV M,A
5938      411E      2E 17 .     MVI L,ZSTAAL*256/256 ;LOW BYTE OF ADDR
5939      4120      C3 2F 41    JMP ALT100
5940      4123      . . .      CTLALT EQU $ ;DEVICE CONTROL
5941      4123      2E 14 .     MVI L,ZCTLAL*256/256
5942      4125      C3 2F 41    JMP ALT100
5943      4128      . . .      BF2ALT EQU $ ;BUFFER TO DEVICE
5944      4128      2E 11 .     MVI L,ZPUTAL*256/256 ;LOW BYTE OF ADDR
5945      412A      C3 2F 41    JMP ALT100
5946      412D      . . .      ALT2BF EQU $ ;DEVICE TO BUFFER
5947      412D      2E 0E .     MVI L,ZGETAL*256/256 ;LOW BYTE OF ADDR
5948      412F      . . .      ALT100 EQU $
5949      412F      26 92 .     MVI H,ALTORG/256 ;HIGH BYTE
5950      4131      CD A5 00    CALL IORMGU ;EXECUTE CODE IF THERE
5951      4134      D0 . .     RNC ;SUCCESS - QUIT
5952      4135      3A 4F FF    LDA IOCERR ;UNSUCCESSFUL - CHECK IOCERR
5953      4138      C3 84 43    JMP ALT500
=====
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 175
5955	413B	.	.	;	
5956	413B	.	.	; DEVICE DRIVERS ENTRY TABLES	
5957	413B	.	.	;	
5958	413B	.	.	; BUFFER TO DEVICE	
5959	413B	.	.	;	
5960	4139	.	.	BF2DTB EQU \$-2	
5961	413B	F0	2F	DW BF2LCT ;LEFT CARTRIDGE	
5962	413D	F5	2F	DW BF2RCT ;RIGHT CARTRIDGE	
5963	413F	C4	43	DW BF2DSP ;DISPLAY	
5964	4141	24	46	DW BF2PRT ;PRINTER	
5965	4143	28	41	DW BF2ALT ;ALTERNATE I/O	
5966	4145	.	.	;	
5967	4145	.	.	; DEVICE TO BUFFER	
5968	4145	.	.	;	
5969	4143	.	.	D2BFTB EQU \$-2	
5970	4145	0B	2E	DW LCT2BF ;LEFT CARTRIDGE	
5971	4147	10	2E	DW RCT2BF ;RIGHT CARTRIDGE	
5972	4149	B5	44	DW DSP2BF ;DISPLAY	
5973	414B	49	47	DW PTR700 ;PRINTER - INVALID	
5974	414D	2D	41	DW ALT2BF ;ALTERNATE I/O	
5975	414F	.	.	;	
5976	414F	.	.	; DEVICE CONTROL	
5977	414F	.	.	;	
5978	414D	.	.	CTLTAB EQU \$-2	
5979	414F	E8	33	DW CTLLCT ;LEFT CARTRIDGE	
5980	4151	FD	33	DW CTRLCT ;RIGHT CARTRIDGE	
5981	4153	63	41	DW NOFNCT ;DISPLAY	
5982	4155	AF	47	DW CTLPRT ;PRINTER	
5983	4157	23	41	DW CTLALT ;ALTERNATE I/O	
5984	4159	.	.	;	
5985	4159	.	.	; DEVICE STATUS	
5986	4159	.	.	;	
5987	4157	.	.	STATTB EQU \$-2	
5988	4159	70	33	DW STLCT ;LEFT CARTRIDGE	
5989	415B	77	33	DW STRCT ;RIGHT CARTRIDGE	
5990	415D	08	46	DW IOSTX1 ;DISPLAY - CLEAR STATUS FLAG	
5991	415F	55	47	DW STPRT ;PRINTER	
5992	4161	15	41	DW STALT ;ALTERNATE I/O	
5993	4163	.	.	;	
5994	4163	.	.	;	
5995	4163	C9	.	NOFNCT RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 176
5997	4164	.	.	.	;
5998	4164	.	.	.	; * * * * *
5999	4164	.	.	.	;
6000	4164	.	.	.	; GETIO - GET NEXT RECORD FROM SOURCE DEVICE
6001	4164	.	.	.	;
6002	4164	.	.	.	; ENTRY GETIO1 REQ'S A = INPUT DEVICE
6003	4164	.	.	.	;
6004	4164	.	.	.	; EXIT : C => ERROR
6005	4164	.	.	.	; NC => NO ERROR
6006	4164	.	.	.	; D,E -> BUFFER STATUS
6007	4164	.	.	.	; A,B,C,H,L DESTROYED
6008	4164	.	.	.	;
6009	4164	.	.	.	GETIO EQU \$
6010	4164	3A	4E	FF	LDA INPDEV ;DEVICE INPUT FLAG => INDEX
6011	4167	.	.	.	GETIO1 EQU \$
6012	4167	47	.	.	MOV B,A ;SAVE DEVICE
6013	4168	21	78	3C	LXI H,CONMSG ;(PREPARE TO REPORT ANY ER
6014	4168	3A	6E	FF	LDA DFLGS ;WRITING TO DISPLAY FROM BUF
6015	416E	E6	80	.	ANI XBF2DS
6016	4170	C2	E5	2D	JNZ CTUERR ;YES - REPORT CONFLICTING I/
6017	4173	78	.	.	MOV A,B ;RECALL DEVICE
6018	4174	21	43	41	LXI H,D2BFTB ;SET TABLE BASE ADDRESS
6019	4177	C3	94	41	JMP SETJMP ;PERFORM INDIRECT JUMP

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
6021     417A      . . .      ;
6022     417A      . . .      ; * * * * *
6023     417A      . . .      ;
6024     417A      . . .      ; PUTIO - PUT RECORD ON ALL "TO" DEVICES
6025     417A      . . .      ;
6026     417A      . . .      ; ENTRY: D,E -> BUFFER STATUS
6027     417A      . . .      ;          BUFFER STATUS, TYPE, LENGTH FILLED
6028     417A      . . .      ;
6029     417A      . . .      ;          ENTRY PUTIO1 REQ'S A = OUTPUT DEVICE(S)
6030     417A      . . .      ;
6031     417A      . . .      ; EXIT : A,B,C,H,L DESTROYED
6032     417A      . . .      ;          NC => NO ERRORS
6033     417A      . . .      ;          D,E -> SAME BUFFER STATUS
6034     417A      . . .      ;          C => ERROR
6035     417A      . . .      ;          D,E DESTROYED
6036     417A      . . .      ;          IOCERR=F => FAILURE
6037     417A      . . .      ;          MSGPTX -> ERROR MESSAGE
6038     417A      . . .      ;          IOCERR=U => USER INTRRUPT
6039     417A      . . .      ;
6040     417A      . . .      ;
6041     417A      . . .      ; PUTIO EQU $
6042     417A      3A 4D FF      ; LDA OUTDEV ;NORMAL OUTPUT DEVICE(S)
6043     417D      . . .      ; PUTIO1 EQU $
6044     417D      12 . .      ; STAX D ;MARK BUFFER FOR OUTPUT
6045     417E      4F . .      ; MOV C,A ;SAVE DEVICE(S)
6046     417F      06 01 . .      ; MVI B,1 ;START WITH UNIT 1 = LFTCTU
6047     4181      . . .      ; PIO010 EQU $
6048     4181      21 39 41      ; LXI H,BF2DTB ;POINTER TO TRANSFER TABLE
6049     4184      79 . .      ; MOV A,C ;THIS DEVICE SELECTED?
6050     4185      A0 . .      ; ANA B
6051     4186      C5 . .      ; PUSH B ;(SAVE CURRENT DEVICE FLAG)
6052     4187      C4 94 41      ; CNZ SETJMP ;YES - PERFORM OUTPUT
6053     418A      C1 . .      ; POP B ;RECALL CURRENT DEVICE
6054     418B      D8 . .      ; RC ;RETURN ON I/O ERROR
6055     418C      78 . .      ; MOV A,B
6056     418D      07 . .      ; RLC ;SELECT NEXT DEVICE
6057     418E      3F . .      ; CMC ;(GET CARRY SET RIGHT)
6058     418F      D0 . .      ; RNC ;RETURN IF FINISHED
6059     4190      47 . .      ; MOV B,A ;SAVE DEVICE FLAG
6060     4191      C3 81 41      ; JMP PIO010 ;GO DO NEXT DEVICE
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 178
=====
6062     4194     . . . ;
6063     4194     . . . ; * * * * *
6064     4194     . . . ;
6065     4194     . . . ; SETJMP - SET UP INDEXED INDIRECT JUMP
6066     4194     . . . ;
6067     4194     . . . ; ENTRY: A = JUMP INDEX (BIT POSITION)
6068     4194     . . . ; E = LSB OF TABLE BASE ADDRESS
6069     4194     . . . ;
6070     4194     . . . ; EXITS TO INDEXED ADDRESS
6071     4194     . . . ;
6072     4194     . . . ;
6073     4194     . . . ; ON ENTRY TO INDEXED ROUTINE, H = BASEH
6074     4194     . . . ; ALL OTHER REGISTERS UNDEFINED
6075     4194     . . . ;
6076     4194     . . . SETJMP EQU $
6077     4194     B7 . . . ORA A ;JUMP TO BASE ?
6078     4195     06 00 . MVI B,0 ;(SET BASE VALUE)
6079     4197     C4 A6 41 . CNZ BT2NUM ;NO - CONVERT BIT TO NUMBER
6080     419A     . . . ;
6081     419A     . . . ; INDJMP - INDEXED INDIRECT JUMP
6082     419A     . . . ;
6083     419A     . . . ; ENTRY: A = INDEX
6084     419A     . . . ; H,L = TABLE BASE ADDRESS
6085     419A     . . . ;
6086     419A     . . . ; EXIT : B = 1
6087     419A     . . . ; A,C DESTROYED
6088     419A     . . . ;
6089     419A     . . . ;
6090     419A     . . . INDJMP EQU $
6091     419A     87 . . . ADD A ;DOUBLE INDEX
6092     419B     4F . . . MOV C,A
6093     419C     06 00 . MVI B,0 ;B,C = INDEX
6094     419E     09 . . . DAD B ;ADD INDEX TO BASE
6095     419F     7E . . . MOV A,M ;GET ADDRESS FROM TABLE
6096     41A0     23 . . . INX H
6097     41A1     66 . . . MOV H,M ;GET HIGH ADDRESS
6098     41A2     6F . . . MOV L,A ;L <- LOW ADDRESS
6099     41A3     06 01 . MVI B,1 ;PUT INDICATOR IN B
6100     41A5     E9 . . . PCHL ;GO THERE
6101     41A6     . . . ;
6102     41A6     . . . ; * * * * *
6103     41A6     . . . ; BT2NUM - CONVERT BIT NUMBER TO DIGIT
6104     41A6     . . . ;
6105     41A6     . . . ; ENTRY: A = BIT TO BE LOCATED
6106     41A6     . . . ; B = BASE VALUE
6107     41A6     . . . ;
6108     41A6     . . . ; EXIT : A,B = BASE VALUE + BIT NUMBER
6109     41A6     . . . ;
6110     41A6     . . . BT2NUM EQU $
6111     41A6     04 . . . INR B ;INCREMENT COUNT
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 179
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 179
6112	41A7	0F . .	RRC ;SHIFT TO NEXT BIT	
6113	41A8	D2 A6 41	JNC BT2NUM ;CONTINUE COUNT IF NO BIT	
6114	41AB	78 . .	MOV A,B	
6115	41AC	C9 . .	RET	

```
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 180
=====
6117     41AD      . . .      ;
6118     41AD      . . .      ; * * * * *
6119     41AD      . . .      ;
6120     41AD      . . .      ; IOFAIL - ERROR IN ESCAPE SEQUENCE
6121     41AD      . . .      ;
6122     41AD      . . .      ;         EXITS TO "ESCEND" THRU "IOEXIT"
6123     41AD      . . .      ;
6124     41AD      . . .      IOFAI0 EQU $
6125     41AD      21 77 3B    LXI H,INOMSG ;SET IN=OUT MESSAGE
6126     41B0      . . .      IOFAI1 EQU $
6127     41B0      22 F1 FF    SHLD MSGPT1
6128     41B3      . . .      ;
6129     41B3      . . .      IOFAIL EQU $
6130     41B3      3E 46 .     MVI A,F
6131     41B5      32 4F FF    STA IOCERR
6132     41B8      37 . .      STC          ;RETURN C => ERROR
6133     41B9      C9 . .      RET
=====
  
```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 181
6135	41BA	. . .	;	
6136	41BA	. . .	;	*****
6137	41BA	. . .	;	
6138	41BA	. . .	;	IORDGO - TRANSFER RECORD TO DATACOM
6139	41BA	. . .	;	
6140	41BA	. . .	;	ENTRY: HANDSHAKE COMPLETED
6141	41BA	. . .	;	IOCTYP = TRANSMISSION TYPE
6142	41BA	. . .	;	0 = ASCII, NEXT BLOCK
6143	41BA	. . .	;	1 = ASCII, LAST BLOCK
6144	41BA	. . .	;	2 = BINARY, NEXT BLOCK
6145	41BA	. . .	;	3 = BINARY, LAST BLOCK
6146	41BA	. . .	;	LSTRED -> START OF LAST BLOCK
6147	41BA	. . .	;	(0 => NO LAST BLOCK)
6148	41BA	. . .	;	NXTRED -> START OF NEXT BLOCK
6149	41BA	. . .	;	(LSB=0 => GET NEW BUFFER FULL)
6150	41BA	. . .	;	NOTE: ASCII XFR IS 1 FIELD (FORMAT
6151	41BA	. . .	;	RECORD) OR 1 NORMAL RECORD.
6152	41BA	. . .	;	BINARY XFR IS ALWAYS 1 RECORD
6153	41BA	. . .	;	BLOCKS STORED IN I/O BUFFERS.
6154	41BA	. . .	;	
6155	41BA	. . .	;	EXIT : LSTRED, NXTRED UPDATED IF NEXT BLK
6156	41BA	. . .	;	WAS REQUESTED
6157	41BA	. . .	;	ASCII XFR - BLOCK SENT
6158	41BA	. . .	;	BINARY XFR - BYTE COUNT SENT,
6159	41BA	. . .	;	MFLGS2[SBINRY] = 1
6160	41BA	. . .	;	
6161	41BA	. . .	;	
6162	41BA	. . .	;	IORDGO EQU \$
6163	41BA	01 FE FF	LXI B,-1-SDVREC ;CLEAR RECORD PENDING FLA	
6164	41BD	CD 55 00	CALL CLBLXF	
6165	41C0	3A D8 FF	LDA IOCTYP ;GET TRANSMISSION TYPE	
6166	41C3	E6 01 .	ANI REXMIT ;RETRANSMIT LAST BLOCK?	
6167	41C5	CA D3 41	JZ IOR020 ;NO - GET NEXT READ POINTER	
6168	41C8	2A 25 FF	LHLD LSTRED ;YES - GET LAST READ POINTER	
6169	41CB	7C . .	MOV A,H ;IS POINTER = 0?	
6170	41CC	B5 . .	ORA L ;(YES => NO LAST BLOCK)	
6171	41CD	CA 45 43	JZ SDEOF ;YES - OUTPUT "FILE MARK"	
6172	41D0	C3 F9 41	JMP IOR100 ;NO - OUTPUT LAST BLOCK	
6173	41D3	. . .	;	*****
6174	41D3	. . .	;	SET UP TO OUTPUT NEW BLOCK *
6175	41D3	. . .	;	*****
6176	41D3	. . .	;	IOR020 EQU \$
6177	41D3	2A 27 FF	LHLD NXTRED ;GET NEXT POINTER	
6178	41D6	B5 . .	ORA L ;LSB = 0?	
6179	41D7	C2 F6 41	JNZ IOR030 ;NO - OUTPUT NEXT FIELD OF	
6180	41DA	. . .	;	CURRENT RECORD
6181	41DA	CD A3 44	CALL INTDS0 ;INPUT=DISPLAY => INITIALIZE	
6182	41DD	3A 26 FF	LDA LSTRED+1 ;GET POINTER TO STATUS OF	
6183	41E0	FE FC .	CPI IOBUF1/256 ;LAST BUFFER OUTPUT	
6184	41E2	11 3A FF	LXI D,B1STAT	

13255

2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
6185     41E5     CA EA 41          JZ   IOR025
6186     41E8     1E 37 .          MVI  E,B2STAT*256/256
6187     41EA     .  .  .          IOR025 EQU $
6188     41EA     CD 64 41        CALL GETIO      ;GET A NEW RECORD FROM THE
6189     41ED     .  .  .          ;          INPUT DEVICE
6190     41ED     DA F0 47        JC   ERRCHK
6191     41F0     3E 20 .          MVI  A,DATCOM  ;MARK FOR OUTPUT TO DATACOM
6192     41F2     12 .  .          STAX D
6193     41F3     CD 2A 3D        CALL GETPTR     ;GET POINTER TO FIRST BYTE
6194     41F6     .  .  .          IOR030 EQU $
6195     41F6     22 25 FF        SHLD LSTRED    ;STORE POINTER FOR REPEAT
6196     41F9     .  .  .          IOR100 EQU $      ;GET TYPE POINTER FOR RECORD
6197     41F9     7C .  .          MOV  A,H
6198     41FA     FE FC .          CPI  IOBUF1/256
6199     41FC     11 39 FF        LXI  D,B1TYPE
6200     41FF     CA 04 42        JZ   IOR110
6201     4202     1E 36 .          MVI  E,B2TYPE*256/256
6202     4204     .  .  .          IOR110 EQU $
6203     4204     1A .  .          LDAX D          ;TYPE OF RECORD?
6204     4205     3C .  .          INR  A          ;(-1 => DATA)
6205     4206     C2 45 43        JNZ  SDOEF      ;TERMINATOR - SEND FILE MARK
6206     4209     3A D8 FF        LDA  IOCTYP     ;GET TYPE OF TRANSMIT
6207     420C     E6 02 .          ANI  BINXMT     ;BINARY TRANSMIT?
6208     420E     C2 CC 42        JNZ  SDBYCT     ;YES - SEND BYTE COUNT
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 183
6210	4211	.	.	*****	
6211	4211	.	.	; SEND ASCII BLOCK TO DATACOM *	
6212	4211	.	.	*****	
6213	4211	45	.	MOV B,L ;B <- # OF BYTES ALREADY SEN	
6214	4212	13	.	INX D ;SAVE POINTER TO BUF STATUS	
6215	4213	D5	.	PUSH D	
6216	4214	1B	.	DCX D ;D,E -> TYPE	
6217	4215	21	6E	FF LXI H,DFLGS ;SET BUFFER TO DISPLAY BIT	
6218	4218	7E	.	MOV A,M	
6219	4219	F6	80	. ORI XBF2DS	
6220	4218	77	.	MOV M,A	
6221	421C	.	.	*****	
6222	421C	.	.	; DETERMINE WHETHER TO SEND INITIAL LINE FEED *	
6223	421C	.	.	*****	
6224	421C	3A	65	FF LDA IOFLGS ;SEND LINE FEED FIRST IF...	
6225	421F	E6	02	. ANI USREAD ;USER READ	
6226	4221	CA	2C	42 JZ IOR210 ;(IOFLGS[USREAD] = 1)	
6227	4224	3A	FB	FF LDA KBJMPR ;AND	
6228	4227	E6	10	. ANI LFPOS ;STRAP 'E' INSERTED	
6229	4229	.	.	. ; (KBJMPR[LFPOS] = 0)	
6230	4229	CC	D7	47 CZ SDAULF ;AND AUTO LF DEPRESSED	
6231	422C	.	.	. ; (MDFLG2[AUTOLF] = 1)	
6232	422C	.	.	. *****	
6233	422C	.	.	. ; OUTPUT A BLOCK TO DATACOM (AND DISPLAY) *	
6234	422C	.	.	. *****	
6235	422C	.	.	. IOR210 EQU \$	
6236	422C	0E	FF	. MVI C,-1 ;C = -1 => PASS CONTROL CODE	
6237	422E	21	B2	42 LXI H,IOR500 ;GET POINTER TO ROUTINE	
6238	4231	97	.	. SUB A ;Z => STRIP OFF TERM CRLF	
6239	4232	CD	39	44 CALL EXPBF1 ;OUTPUT THE BUFFER, EXPANDIN	
6240	4235	.	.	. ; CONTROL CODES	
6241	4235	D2	3E	42 JNC IOR230 ;CONTINUE ON NO ERROR	
6242	4238	.	.	. *****	
6243	4238	.	.	. ; ERROR - ABORT READ *	
6244	4238	.	.	. *****	
6245	4238	.	.	. IOR220 EQU \$	
6246	4238	D1	.	. POP D ;D,E -> BUF STATUS	
6247	4239	97	.	. SUB A ;FREE BUFFER	
6248	423A	12	.	. STAX D	
6249	423B	C3	75	43 JMP RDABRT ;ABORT READ	
6250	423E	.	.	. IOR230 EQU \$;INTERPRET EXPBUF RETURN	
6251	423E	B7	.	. ORA A ;TYPE OF TERMINATION?	
6252	423F	C2	50	42 JNZ IOR270 ;FORMAT FIELD SEPARATOR -	
6253	4242	E3	.	. XTHL ;NEXT-TO-LAST CHAR	
6254	4243	7E	.	. MOV A,M ;MARK BUF OK FOR PTTPLN	
6255	4244	E6	DF	. ANI -1-DATCOM	
6256	4246	F6	40	. ORI PTTPOK	
6257	4248	77	.	. MOV M,A	
6258	4249	E3	.	. XTHL ;RESTORE H,L -> ROUTINE	
6259	424A	CD	6B	44 CALL EXPBF3 ;OUTPUT THE LAST CHARACTER	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 184
=====
6260     424D     C3  59  42          JMP  IOR285      ;GO TO CLEAN UP ROUTINE
6261     4250     .   .   .          ;*****
6262     4250     .   .   .          ; HANDLE FORMAT MODE SEPARATORS (304B) *
6263     4250     .   .   .          ;*****
6264     4250     .   .   .          IOR270 EQU $
6265     4250     13  .   .          INX  D          ;SKIP FORMAT COUNT
6266     4251     05  .   .          DCR  B
6267     4252     78  .   .          MOV  A,E        ;IS THIS PART OF HEADER?
6268     4253     FE  05  .          CPI  5          ;(FIRST FOUR BYTES)
6269     4255     3D  .   .          DCR  A          ;(INSURE NC, NZ IF NOT)
6270     4256     DC  67  44          CC   EXPBF2     ;YES - CONTINUE OUTPUTTING
6271     4259     .   .   .          IOR285 EQU $
6272     4259     DA  38  42          JC   IOR220     ;ABORT ON ERROR
6273     425C     CA  3E  42          JZ   IOR230     ;CONTINUE ON BUF NOT EMPTY
6274     425F     EB  .   .          XCHG          ;END OF BLOCK
6275     4260     D1  .   .          POP  D          ;RECALL TYPE POINTER
6276     4261     7D  .   .          MOV  A,L        ;BUFFER EXHAUSTED?
6277     4262     B7  .   .          ORA  A
6278     4263     C2  67  42          JNZ  IOR300     ;NO - SEND TERMINATOR
6279     4266     12  .   .          STAX D
=====
  
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
6281      4267      . . .      ;
6282      4267      . . .      ;*****
6283      4267      . . .      ;  TERMINATE THE READ BY SENDING THE FOLLOWING:  *
6284      4267      . . .      ;
6285      4267      . . .      ;      USER READ, STRAP E IN:      CR      *
6286      4267      . . .      ;      USER READ, STRAP E OUT:     CR(LF)  *
6287      4267      . . .      ;      REMOTE READ, BLOCK, PAGE:  CR(LF)RS *
6288      4267      . . .      ;      REMOTE READ, OTHER:        CR(LF)  *
6289      4267      . . .      ;
6290      4267      . . .      ;      IF USER READ, HALF DUPLEX, NON-FORMAT MODE,  *
6291      4267      . . .      ;      SEND CR(LF) TO DISPLAY      *
6292      4267      . . .      ;*****
6293      4267      . . .      IOR300 EQU $
6294      4267      22 27 FF      SHLD NXTRED      ;SAVE POINTER FOR NEXT READ
6295      426A      3A 65 FF      LDA IOFLGS      ;USER READ?
6296      426D      E6 02 .      ANI USREAD
6297      426F      CA 83 42      JZ IOR350      ;NO - TERMINATE REMOTE READ
6298      4272      3A FB FF      LDA KBJMPR      ;STRAP 'E' IN?
6299      4275      2F . .      CMA      ;(COMPLEMENT FLAGS)
6300      4276      E6 10 .      ANI LFPOS      ;LINE FEED AT START OF REC?
6301      4278      3E 0D .      MVI A,CR      ;(SET TO OUTPUT RETURN)
6302      427A      C2 8C 42      JNZ IOR360      ;YES - OUTPUT RETURN ONLY
6303      427D      CD D2 47      CALL SDCRLF      ;NO - OUTPUT CR(LF)
6304      4280      C3 8F 42      JMP IOR380      ;TERMINATE OUTPUT BLOCK
6305      4283      . . .      ;*****
6306      4283      . . .      ;  TERMINATE REMOTE READ *
6307      4283      . . .      ;*****
6308      4283      . . .      IOR350 EQU $
6309      4283      CD D2 47      CALL SDCRLF      ;SEND CR(LF)
6310      4286      CD B1 00      CALL GTMODE      ;PAGE MODE?
6311      4289      3A 04 50      LDA BLKTRM      ;(SET TO OUTPUT TERM CHAR)
6312      428C      . . .      IOR360 EQU $
6313      428C      C4 A7 42      CNZ IOR400      ;YES - OUTPUT TERMINATOR
6314      428F      . . .      IOR380 EQU $
6315      428F      CD AC 43      CALL CLXB2D      ;CLEAR BUFFER TO DISPLAY BIT
6316      4292      2E 65 .      MVI L,IOFLGS
6317      4294      7E . .      MOV A,M
6318      4295      E6 01 .      ANI RDWOWT      ;READ WITHOUT WAIT?
6319      4297      C2 BA 41      JNZ IORDGO      ;YES - START ANOTHER READ
6320      429A      CD F2 3C      CALL ENDATA      ;NO - SIGNAL END OF DATA BLO
6321      429D      01 01 00      LXI B,SDVREC      ;SET TO ENABLE ANOTHER READ
6322      42A0      7E . .      MOV A,M      ;GET IOFLGS
6323      42A1      E6 06 .      ANI USREAD+FILRED ;USER/FILE READ?
6324      42A3      C2 58 00      JNZ SBLXF0      ;YES - ENABLE ANOTHER READ
6325      42A6      C9 . .      RET      ;RETURN
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
6327      42A7      . . . ;
6328      42A7      . . . ; * * * * *
6329      42A7      . . . ;
6330      42A7      . . . ; GENERAL OUTPUT TO DATACOM ROUTINE
6331      42A7      . . . ;
6332      42A7      . . . ; ECHOS TO DISPLAY IF USER READ, HALF DUPLEX
6333      42A7      . . . ;
6334      42A7      . . . ; ENTRY IOR400, FOR READ TERMINATORS,
6335      42A7      . . . ; SUPRESSES ECHO IF IN FORMAT MODE
6336      42A7      . . . ;
6337      42A7      . . . ; ENTRY:  A = CHAR
6338      42A7      . . . ;
6339      42A7      . . . ; EXIT :  A,H,L DESTROYED
6340      42A7      . . . ; NC => NO ERROR
6341      42A7      . . . ; C => ERROR OCCURRED
6342      42A7      . . . ;
6343      42A7      . . . IOR400 EQU $
6344      42A7      6F . . . MOV L,A ;SAVE A-REGISTER
6345      42A8      CD CC 47 CALL CHKFMT ;FORMAT MODE ENABLED?
6346      42AB      7D . . . MOV A,L ;(RECALL A-REGISTER)
6347      42AC      CA B5 42 JZ IOR550 ;NO - DON'T SUPPRESS ECHO
6348      42AF      C3 7C 00 JMP XPUTD3 ;YES - OUTPUT TERMINATOR ONL
6349      42B2      . . . IOR500 EQU $
6350      42B2      FE 0A . . . CPI LF ;STRIP OUT LINE FEEDS
6351      42B4      C8 . . . RZ
6352      42B5      . . . IOR550 EQU $
6353      42B5      C5 . . . PUSH B
6354      42B6      D5 . . . PUSH D
6355      42B7      4F . . . MOV C,A ;CHINT WANTS BYTE IN C
6356      42B8      CD 7C 00 CALL XPUTD3 ;OUTPUT TO DATACOM
6357      42B8      DA C9 42 JC IOR570 ;QUIT ON ERROR
6358      42BE      3A 64 FF LDA IOFLG2 ;USER READ, HALF DUP?
6359      42C1      E6 01 . . . ANI EXT82D
6360      42C3      CC 8B 43 CZ XCHINT ;YES - SEND TO DISPLAY
6361      42C6      D4 94 00 CNC GETDCM ;IF NO ERROR, MONIT DATACOM
6362      42C9      . . . IOR570 EQU $
6363      42C9      D1 . . . POP D
6364      42CA      C1 . . . POP B
6365      42CB      C9 . . . RET
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 187
6367	42CC	. . .	;	
6368	42CC	. . .	; * * * * *	
6369	42CC	. . .	;	
6370	42CC	. . .	; SDBYCT - SEND BYTE COUNT TO DATACOM	
6371	42CC	. . .	;	
6372	42CC	. . .	; FIRST PART OF BINARY READ--ENTERED FROM	
6373	42CC	. . .	; IORDGO (HANDSHAKE COMPLETED).	
6374	42CC	. . .	; RETURNS TO WAIT LOOP, HAVING SET	
6375	42CC	. . .	; MFLGS2[SBINRY], WHICH TRIGGERS CALL TO	
6376	42CC	. . .	;	
6377	42CC	. . .	; ENTRY: D,E -> RECORD TYPE (IORDGO HANDLES	
6378	42CC	. . .	; NON-DATA RECORDS)	
6379	42CC	. . .	; H,L -> FIRST BYTE OF RECORD	
6380	42CC	. . .	; NC (REQ'D BY FIRST CALL TO XPUTDC)	
6381	42CC	. . .	;	
6382	42CC	. . .	; EXIT : 4-BYTE COUNT AND TERMINATOR SENT	
6383	42CC	. . .	; MFLGS2[SBINRY] = 1	
6384	42CC	. . .	; NXTRED[0-4]=0	
6385	42CC	. . .	; (= > NEXT READ WILL GET NEW REC)	
6386	42CC	. . .	;	
6387	42CC	. . .	;	
6388	42CC	. . .	SDBYCT EQU \$	
6389	42CC	97 . .	SUB A ;CLEAR NXTRED[0-4]	
6390	42CD	32 27 FF	STA NXTRED	
6391	42D0	3E 30 .	MVI A,60Q ;SEND LEADING '0'	
6392	42D2	CD 7C 00	CALL XPUTD3 ;OUTPUT 1ST BYTE	
6393	42D5	1B . .	DCX D ;D,E -> BYTE COUNT	
6394	42D6	1A . .	LDAX D	
6395	42D7	95 . .	SUB L ;SUBTRACT BYTES ALREADY SENT	
6396	42D8	. . .	; (BY ASCII READ OF FORMAT	
6397	42D8	. . .	; RECORD)	
6398	42D8	47 . .	MOV B,A ;SAVE BYTE COUNT	
6399	42D9	D6 01 .	SUI 1 ;IS COUNT = 0 (= > 256)?	
6400	42DB	3E 30 .	MVI A,60Q ;IF NO (NC) - 2ND BYTE = '0'	
6401	42D0	CE 00 .	ACI 0 ;IF YES (C) - 2ND BYTE = '1'	
6402	42DF	CD 7C 00	CALL XPUTD3 ;OUTPUT 2ND BYTE	
6403	42E2	78 . .	MOV A,B ;GET LENGTH	
6404	42E3	E6 F0 .	ANI 360Q ;EXTRACT HIGH 4 BITS	
6405	42E5	0F . .	RRC ;RIGHT JUSTIFY	
6406	42E6	0F . .	RRC	
6407	42E7	0F . .	RRC	
6408	42E8	0F . .	RRC	
6409	42E9	F6 30 .	ORI 60Q ;MAKE IT AN ASCII CHAR	
6410	42EB	CD 7C 00	CALL XPUTD3 ;OUTPUT 3RD BYTE	
6411	42EE	78 . .	MOV A,B ;GET LENGTH	
6412	42EF	E6 0F .	ANI 17Q ;EXTRACT LOW 4 BITS	
6413	42F1	F6 30 .	ORI 60Q ;MAKE IT AN ASCII CHAR	
6414	42F3	CD 7C 00	CALL XPUTD3 ;OUTPUT 4TH BYTE	
6415	42F6	D4 79 00	CNC SDTRM1 ;OUTPUT TERMINATOR	
6416	42F9	DA 75 43	JC RDABRT ;ABORT READ IF OUTPUT FAILED	

13255

2648A MICROCODE LISTING 'I0273'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 188
=====
```

6417	42FC	3A 65 FF	LDA IOFLGS ;GET I/O FLAGS
6418	42FF	E6 01 .	ANI RDWOWT ;READ WITHOUT WAIT ENABLED?
6419	4301	C2 0D 43	JNZ BNRYGO ;YES - SEND BINARY NOW
6420	4304	CD F2 3C	CALL ENDATA ;NO - SIGNAL END OF DATA BLO
6421	4307	01 02 00	LXI B,SBINRY ;SET BINARY RECORD PENDING
6422	430A	C3 58 00	JMP SBLXF0

```
=====
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 189
6424	430D	.	.	.	;
6425	430D	.	.	.	; * * * * *
6426	430D	.	.	.	;
6427	430D	.	.	.	; BNR005 - SEND BINARY RECORD
6428	430D	.	.	.	;
6429	430D	.	.	.	; ENTRY: BYTE COUNT SENT, 2ND HANDSHAKE DONE
6430	430D	.	.	.	; LSTRED -> 1ST BYTE OF RECORD
6431	430D	.	.	.	;
6432	430D	.	.	.	; EXIT: RECORD SENT
6433	430D	.	.	.	; BUFFER RELEASED
6434	430D	.	.	.	;
6435	430D	.	.	.	;
6436	430D	.	.	.	; BNR005 EQU \$
6437	430D	2A	25	FF	LHLD LSTRED ;GET POINTER TO FIRST BYTE
6438	4310	7C	.	.	MOV A,H ;GET POINTER TO LENGTH
6439	4311	FE	FC	.	CPI IOBUF1/256 ;DATA IN BUFFER 1?
6440	4313	11	38	FF	LXI D,B1LEN ;(SET FOR BUFFER 1 LENGTH)
6441	4316	CA	1B	43	JZ BNR005 ;YES - OUTPUT THE BUFFER
6442	4319	1E	35	.	MVI E,B2LEN*256/256
6443	431B	.	.	.	; BNR005 EQU \$
6444	431B	CD	20	50	CALL ZSTBIN ;SIGNAL START OF BINARY OUTP
6445	431E	CD	2A	43	CALL BNR010 ;OUTPUT THE BINARY RECORD
6446	4321	DA	72	43	JC BNAVRT ;ABORT BINARY IF FAILURE
6447	4324	CD	23	50	CALL ZNDBIN ;END BINARY OUTPUT MODE
6448	4327	C3	8F	42	JMP IOR380 ;GO TO READ EXIT ROUTINE
6449	432A	.	.	.	;*****
6450	432A	.	.	.	; BNR010 - OUTPUT BINARY DATA *
6451	432A	.	.	.	;*****
6452	432A	.	.	.	; BNR010 EQU \$
6453	432A	E5	.	.	PUSH H ;SAVE FIRST BYTE POINTER
6454	432B	01	FC	00	LXI B,377Q-SBINRY-SDVREC
6455	432E	CD	55	00	CALL CLBLXF ;CLEAR ALL PENDING XFERS
6456	4331	E1	.	.	POP H ;RECALL FIRST BYTE POINTER
6457	4332	1A	.	.	LDAX D ;GET RECORD LENGTH
6458	4333	95	.	.	SUB L ;SUBTRACT BYTES ALREADY SENT
6459	4334	D8	.	.	RC ;C => BUFFER OVERWRITTEN
6460	4335	47	.	.	MOV B,A ;SAVE # OF BYTES LEFT
6461	4336	.	.	.	; BNR020 EQU \$
6462	4336	7E	.	.	MOV A,M ;GET THE DATA BYTE
6463	4337	CD	7C	00	CALL XPUTD3 ;OUTPUT THE BINARY BYTE
6464	433A	D8	.	.	RC ;RETURN ON DATA COMM ERROR
6465	433B	23	.	.	INX H ;INCREMENT TO NEXT BYTE
6466	433C	05	.	.	DCR B ;ALL BYTES DONE?
6467	433D	C2	36	43	JNZ BNR020 ;NO - CONTINUE TRANSMITTING
6468	4340	13	.	.	INX D ;YES - RELEASE THE BUFFER
6469	4341	13	.	.	INX D
6470	4342	97	.	.	SUB A
6471	4343	12	.	.	STAX D ;CLEAR BUFFER BUSY FLAG
6472	4344	C9	.	.	RET ;RETURN

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 190
=====
6474      4345      . . .      ;
6475      4345      . . .      ; * * * * *
6476      4345      . . .      ;
6477      4345      . . .      ; SDEOF - SEND SPECIAL RECORD AS "READ"
6478      4345      . . .      ; RESPONSE
6479      4345      . . .      ;
6480      4345      . . .      ; SENT FOR END OF FILE, END OF DATA, I/O
6481      4345      . . .      ; ERROR, REPEAT REQUEST WITH NO
6482      4345      . . .      ; PREVIOUS READ.
6483      4345      . . .      ;
6484      4345      . . .      ; CHARACTERS SENT:
6485      4345      . . .      ; USER READ                                CR(LF)
6486      4345      . . .      ; (IF HALF DUPLEX, DISP GETS CR(LF))
6487      4345      . . .      ; REMOTE READ, PAGE, BLOCK RS
6488      4345      . . .      ; REMOTE READ, OTHER RS CR(LF)
6489      4345      . . .      ;
6490      4345      . . .      ; RELEASES BOTH I/O BUFFERS.
6491      4345      . . .      ;
6492      4345      . . .      ;
6493      4345      . . .      SDEOF EQU $
6494      4345      3A 65 FF LDA IOFLGS ;GET I/O FLAGS
6495      4348      2F . . CMA ;COMPLEMENT THE FLAGS
6496      4349      E6 02 . ANI USREAD ;USER INITIATED READ?
6497      4348      CA 57 43 JZ SEF100 ;YES - SEND CR(LF)
6498      434E      3A 04 50 LDA BLKTRM ;NO - SEND BLOCK TERMINATOR
6499      4351      CD 7C 00 CALL XPUTD3 ;CHARACTER
6500      4354      CD B1 00 CALL GTMODE ;BLOCK, PAGE?
6501      4357      . . . SEF100 EQU $
6502      4357      CC D2 47 CZ SDCRLF ;NO - SEND CR(LF)
6503      435A      . . . SDEOF1 EQU $ ;ENTRY FOR RDABRT
6504      435A      CD 17 3D CALL FREBFS ;RELEASE I/O BUFFERS
6505      435D      97 . . SUB A
6506      435E      32 27 FF STA NXTRED ;NEXT READ WILL GET NEW REC
6507      4361      CD F2 3C CALL ENDATA ;SIGNAL END OF DATA BLOCK
6508      4364      3E FA . MVI A,-1-RDWOWT-FILRED
6509      4366      CD 2A 28 CALL CLIOFS ;CLEAR READ FLAGS
6510      4369      3E FD . MVI A,-1-USREAD
6511      4368      CD 2A 28 CALL CLIOFS ;CLEAR USER READ FLAG
6512      436E      C8 . . RZ ;RETURN IF REMOTE READ
6513      436F      C3 84 36 JMP USREXT ;REPORT ANY ERRORS
=====

```


13255

2648A MICROCODE LISTING 'I0273'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS      PAGE 192
=====
```

6565	43AE	21 6E FF	LXI H,DFLGS
6566	43B1	A6 . .	ANA M
6567	43B2	77 . .	MOV M,A
6568	43B3	C9 . .	RET

```
=====
```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 193
=====
6570     43B4      . . .      ALT500 EQU $
6571     43B4      FE 55 .      CPI U      ;FAILURE FROM USER INTERRUPT
6572     43B6      C2 B3 41    JNZ IOFAIL ;NO - MARK IOCERR F
6573     43B9      37 . .      STC
6574     43BA      C9 . .      RET
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 194
=====
6576     43BB      . . .      ;
6577     43BB      . . .      ; * * * * *
6578     43BB      . . .      ;
6579     43BB      . . .      ;      XCHINT - INTERPRET CHAR, AND CHECK FOR
6580     43BB      . . .      ;      MEMORY LOCKUP
6581     43BB      . . .      ;
6582     43BB      . . .      ;      ENTRY:  C = CHAR
6583     43BB      . . .      ;
6584     43BB      . . .      ;      EXIT :  C => ERROR, MEMORY LOCKED
6585     43BB      . . .      ;      NC => NO ERROR
6586     43BB      . . .      ;
6587     43BB      . . .      ;
6588     43BB      . . .      XCHINT EQU $
6589     43BB      CD 82 00    CALL CHINT
6590     43BE      3A 6A FF    LDA MLKFLG      ;MLKFLG <> 0 => LOCKUP
6591     43C1      C6 FF .     ADI 377Q
6592     43C3      C9 . .     RET
  
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 195
6594	43C4	.	.	.	;
6595	43C4	.	.	.	; * * * * *
6596	43C4	.	.	.	;
6597	43C4	.	.	.	; BF2DSP - DISPLAY CONTENTS OF I/O BUFFER
6598	43C4	.	.	.	;
6599	43C4	.	.	.	; ENTRY: D,E -> STATUS
6600	43C4	.	.	.	;
6601	43C4	.	.	.	; EXIT : D,E -> STATUS
6602	43C4	.	.	.	; STATUS BIT FOR DISPLAY CLEARED
6603	43C4	.	.	.	; DESTROYS A,B,C,H,L
6604	43C4	.	.	.	;
6605	43C4	.	.	.	BF2DSP EQU \$
6606	43C4	21	6E	FF	LXI H,DFLGS ;SET BUFFER TO DISPLAY BIT
6607	43C7	7E	.	.	MOV A,M
6608	43C8	F6	80	.	ORI XBF2DS
6609	43CA	77	.	.	MOV M,A
6610	43CB	D5	.	.	PUSH D ;SAVE D,E FOR RETURN
6611	43CC	1B	.	.	DCX D ;D,E -> TYPE
6612	43CD	1A	.	.	LDAX D
6613	43CE	B7	.	.	ORA A ;WHAT TYPE OF RECORD?
6614	43CF	F2	0E	44	JP B2D030 ;EOF OR EVD - QUIT
6615	43D2	0E	FF	.	MVI C,-1 ;C = -1 => SEND CONTROL CODE
6616	43D4	21	19	44	LXI H,B2D100 ;PTR TO ROUTINE FOR EACH BYT
6617	43D7	.	.	.	*****
6618	43D7	.	.	.	; DO NOT STRIP CR OR LF IF IN TEK MODE
6619	43D7	CD	70	60	CALL ZCHKTK ;IN TEK MODE?
6620	43DA	CA	E0	43	JZ B2D003 ;NO, FLAG = Z
6621	43DD	21	1C	44	LXI H,B2D120 ;YES, FLAG = NZ
6622	43E0	.	.	.	B2D003 EQU \$
6623	43E0	.	.	.	*****
6624	43E0	CD	37	44	CALL EXPBF0 ;START OUTPUTTING BUFFER
6625	43E3	DA	0E	44	JC B2D030 ;RETURN ON ERROR
6626	43E6	.	.	.	B2D005 EQU \$
6627	43E6	B7	.	.	ORA A ;TYPE OF RETURN?
6628	43E7	C2	F8	43	JNZ B2D010 ;FORMAT FIELD SEPARATOR -
6629	43EA	E3	.	.	XTHL ;NEXT-TO-LAST CHAR
6630	43EB	7E	.	.	MOV A,M ;SET BUF AVAIL FOR PTTPLN
6631	43EC	E6	FB	.	ANI -1-DISPLY
6632	43EE	F6	40	.	ORI PTTPOK
6633	43F0	77	.	.	MOV M,A
6634	43F1	E3	.	.	XTHL
6635	43F2	CD	6B	44	CALL EXPBF3 ;OUTPUT LAST CHARACTER
6636	43F5	C3	05	44	JMP B2D020 ;OUTPUT LAST CHAR

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 196
=====
6638     43F8      . . .      ;
6639     43F8      . . .      ; HANDLE FORMAT FIELD SEPARATORS
6640     43F8      . . .      ;
6641     43F8      . . .      B2D010 EQU $
6642     43F8      13 . .     INX D ;SKIP FIELD PARAMETER
6643     43F9      05 . .     DCR B
6644     43FA      3E 04 . .   MVI A,4 ;PART OF HEADER?
6645     43FC      BB . .     CMP E ;(1ST FOUR BYTES)
6646     43FD      E5 . .     PUSH H
6647     43FE      DC 25 44    CC B2D200 ;NO - DISPLAY CR LF
6648     4401      E1 . .     POP H
6649     4402      CD 67 44    CALL EXPBF2 ;START NEXT FORMAT FIELD
6650     4405      . . .      B2D020 EQU $
6651     4405      DA 0E 44    JC B2D030 ;RETURN ON ERROR
6652     4408      CA E6 43    JZ B2D005 ;HANDLE ANY MORE FIELD SEP
6653     4408      . . .      ;
6654     4408      . . .      ; DECIDE WHETHER TO DO CR LF
6655     4408      . . .      ;
6656     4408      CD 25 44    CALL B2D200 ;DISPLAY CR LF
6657     440E      . . .      B2D030 EQU $
6658     440E      D1 . .     POP D ;RECALL BUFFER STATUS PTR
6659     440F      F5 . .     PUSH PSW ;SAVE ERROR (C) FLAG
6660     4410      1A . .     LDAX D ;CLEAR DISPLAY BIT IN STATUS
6661     4411      E6 BB . .   ANI -1-DISPLY-PTTPOK
6662     4413      12 . .     STAX D
6663     4414      CD AC 43    CALL CLXB2D ;CLEAR BUFFER TO DISPLAY BIT
6664     4417      F1 . .     POP PSW ;RECALL ERROR (C) FLAG
6665     4418      C9 . .     RET
6666     4419      . . .      ;
6667     4419      . . .      ; ROUTINE HANDLES EACH BYTE
6668     4419      . . .      ;
6669     4419      . . .      B2D100 EQU $
6670     4419      FE 0A . .   CPI LF ;STRIP OUT LINE FEEDS
6671     4418      C8 . .     RZ
6672     441C      . . .      B2D120 EQU $
6673     441C      C5 . .     PUSH B ;CHINT KILLS ALL
6674     441D      D5 . .     PUSH D
6675     441E      4F . .     MOV C,A ;CHINT TAKES INPUT IN C
6676     441F      CD BB 43    CALL XCHINT ;DISPLAY BYTE
6677     4422      D1 . .     POP D
6678     4423      C1 . .     POP B
6679     4424      C9 . .     RET
6680     4425      . . .      ;
6681     4425      . . .      ; IF NON-FORMAT MODE, SEND CR LF TO DISPLAY
6682     4425      . . .      ;
6683     4425      . . .      B2D200 EQU $
6684     4425      CD CC 47    CALL CHKfmt ;FORMAT MODE?
6685     4428      C0 . .     RNZ ;YES - QUIT
6686     4429      . . .      ;*****
6687     4429      . . .      ; DONT APPEND CR/LF IF IN TEK MODE
=====

```



```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 197
=====
6688     4429     CD  70  60          CALL ZCHKTK          ;IN TEK MODE?
6689     442C     C0  .   .          RNZ                  ;YES
6690     442D     .   .   .          ;*****
6691     442D     3E  0D  .          MVI A,CR             ;NO - SEND CR
6692     442F     CD  1C  44          CALL B2D120          ;(SAVING REGISTERS)
6693     4432     3E  0A  .          MVI A,LF             ;AND LF
6694     4434     C3  1C  44          JMP B2D120
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 198
6696	4437	. . .	;	
6697	4437	. . .	; * * * * *	
6698	4437	. . .	;	
6699	4437	. . .	;	
6700	4437	. . .	EXPBUF - PROCESS BUFFER OF DATA,	
6701	4437	. . .	EXPANDING CONTROL CODES	
6702	4437	. . .	;	
6703	4437	. . .	CALLLED BY IORDGO (BUFFER TO DATACOM)	
6704	4437	. . .	BF2DSP (BUFFER TO DISPLAY)	
6705	4437	. . .	BF2PRT (BUFFER TO PRINTER)	
6706	4437	. . .	;	
6707	4437	. . .	INITIALIZATION:	
6708	4437	. . .	ENTRY EXPBF0 - SKIP NO BYTES	
6709	4437	. . .	ENTRY EXPBF1 - SKIP FIRST (B-REG) BYTES	
6710	4437	. . .	Z => DELETE TERMINATING CRLF	
6711	4437	. . .	NZ,NC => DO NOT DELETE	
6712	4437	. . .	C = 0 => SKIP CONTROL CODES	
6713	4437	. . .	C = -1 => PASS CONTROL CODES	
6714	4437	. . .	D,E -> BUF TYPE(MUST BE -1 => DATA)	
6715	4437	. . .	H,L -> SUBROUTINE TO BE EXECUTED	
6716	4437	. . .	FOR EACH BYTE PROCESSED:	
6717	4437	. . .	ENTRY: A = BYTE	
6718	4437	. . .	EXIT : NC => NO ERROR	
6719	4437	. . .	C => FATAL ERROR	
6720	4437	. . .	A,H,L DESTROYED	
6721	4437	. . .	;	
6722	4437	. . .	CONTINUATION - ENTRY EXPBF2:	
6723	4437	. . .	B = BYTES REMAINING	
6724	4437	. . .	C-REG AS ABOVE	
6725	4437	. . .	D,E -> NEXT BYTE TO BE PROCESSED	
6726	4437	. . .	H,L -> SUBROUTINE	
6727	4437	. . .	;	
6728	4437	. . .	EXIT : C => FATAL ERROR	
6729	4437	. . .	A-E DESTROYED	
6730	4437	. . .	NC => NO ERROR	
6731	4437	. . .	NZ => BUFFER EXHAUSTED	
6732	4437	. . .	A-D DESTROYED	
6733	4437	. . .	E = 0	
6734	4437	. . .	Z => FIELD SEPARATOR FOUND	
6735	4437	. . .	A = 304B => FIELD SEPARATOR	
6736	4437	. . .	A = 0 => NEXT-TO-LAST CHAR	
6737	4437	. . .	B = BYTES REMAINING	
6738	4437	. . .	C-REG AS ABOVE	
6739	4437	. . .	D,E -> NXT BYTE TO BE PROCESSED	
6740	4437	. . .	H,L -> SUBROUTINE	
6741	4437	. . .	;	
6742	4437	. . .	A "NEXT-TO-LAST CHAR" RETURN IS GUARANTEED,	
6743	4437	. . .	EVEN IF THE BUFFER IS EMPTY (ONLY CR LF).	
6744	4437	. . .	IN THAT CASE, THE NEXT CALL RETURNS BUFFER	
6745	4437	. . .	EXHAUSTED.	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 199
6746	4437	.	.	.	EXPBF0 EQU \$	
6747	4437	06	00	.	MVI B,0	;B = 0 => SKIP NO BYTES
6748	4439	.	.	.	EXPBF1 EQU \$	
6749	4439	E5	.	.	PUSH H	;SAVE ADDRESS
6750	443A	F5	.	.	PUSH PSW	;SAVE C => DO NOT DEL CRLF
6751	443B	2A	2A	48	LHLD ALTOUT	;CLEAR CURRENT ENHANCEMENT
6752	443E	22	75	FF	SHLD CALTST	;AND ALTERNATE CHAR SET
6753	4441	CD	2A	3D	CALL GETPTR	;GET POINTER TO FIRST BYTE
6754	4444	1B	.	.	DCX D	;D,E -> LENGTH
6755	4445	1A	.	.	LDAX D	;GET LENGTH
6756	4446	54	.	.	MOV D,H	
6757	4447	58	.	.	MOV E,B	;D,E -> FIRST BYTE TO PROCES
6758	4448	6F	.	.	MOV L,A	
6759	4449	2D	.	.	DCR L	;H,L -> LAST BYTE
6760	444A	90	.	.	SUB B	;A = REMAINING BYTES
6761	444B	47	.	.	MOV B,A	;B = COUNTER
6762	444C	F1	.	.	POP PSW	;RECALL Z-BIT
6763	444D	CC	93	44	CZ STCRLF	;IF Z, STRIP CR LF
6764	4450	E1	.	.	POP H	
6765	4451	C8	.	.	RZ	;RETURN IF 0 OR 1 CHARS LEFT
6766	4452	78	.	.	MOV A,B	;ONLY 1 CHAR LEFT ?
6767	4453	EE	01	.	XRI 10	
6768	4455	C8	.	.	RZ	;YES - RET NEXT-TO-LAST CHAR
6769	4456	.	.	.	EXB020 EQU \$	
6770	4456	1A	.	.	LDAX D	;GET NEXT BYTE
6771	4457	13	.	.	INX D	;ADVANCE POINTER
6772	4458	05	.	.	DCR B	;DECREMENT COUNTER
6773	4459	CA	5F	44	JZ EXB030	;LAST CHAR=FIELD SEP=> NO RE
6774	445C	FE	C4	.	CPI FLDSEP	;IS IT A FIELD SEPARATOR?
6775	445E	C8	.	.	RZ	;YES - RETURN
6776	445F	.	.	.	EXB030 EQU \$	
6777	445F	B7	.	.	ORA A	;NO - IS IT A CONTROL CODE?
6778	4460	FA	73	44	JM EXB100	;YES - EXPAND IT
6779	4463	E5	.	.	PUSH H	;NO - SAVE SUBROUTINE ADDRESS
6780	4464	CF	.	.	RST 1	;PERFORM THE SUBROUTINE
6781	4465	E1	.	.	POP H	;RESTORE SUBROUTINE ADDRESS
6782	4466	.	.	.	EXB050 EQU \$;RE-ENTRY FOR ESCAPE SEQUENC
6783	4466	D8	.	.	RC	;RETURN ON ERROR

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 200
=====
6785     4467     . . .      ;*****
6786     4467     . . .      ; CONTINUE AFTER FIELD SEPARATOR *
6787     4467     . . .      ;*****
6788     4467     . . .      EXPBF2 EQU $
6789     4467     78 . . .      MOV A,B          ;ONLY ONE CHAR LEFT?
6790     4468     EE 01 . . .      XRI 1Q
6791     446A     C8 . . .      RZ              ;YES - RET "PENULTIMATE CHAR
6792     446B     . . .      ;*****
6793     446B     . . .      ; CONTINUATION ENTRY *
6794     446B     . . .      ;*****
6795     446B     . . .      EXPBF3 EQU $
6796     446B     78 . . .      MOV A,B          ;ANY CHARS LEFT?
6797     446C     B7 . . .      ORA A
6798     446D     C2 56 44 . . .      JNZ EXB020       ;YES - PROCESS NEXT BYTE
6799     4470     5F . . .      MOV E,A          ;YES - CLEAR E-REG
6800     4471     3C . . .      INR A           ;SET NZ => BUFFER EXHAUSTED
6801     4472     C9 . . .      RET
6802     4473     . . .      ;*****
6803     4473     . . .      ; HANDLE CONTROL CODES *
6804     4473     . . .      ;*****
6805     4473     . . .      EXB100 EQU $
6806     4473     A1 . . .      ANA C           ;PASS CONTROL CODES?
6807     4474     CA 66 44 . . .      JZ EXB050       ;NO - CONTINUE
6808     4477     C5 . . .      PUSH B          ;SAVE REGISTERS
6809     4478     D5 . . .      PUSH D
6810     4479     EB . . .      XCHG
6811     447A     4F . . .      MOV C,A         ;EXPAND WANTS BYTE IN C & A
6812     447B     CD 8E 00 . . .      CALL EXPAND     ;EXPAND TO ESC SEQUENCE
6813     447E     EB . . .      XCHG           ;H,L -> SUBROUTINE
6814     447F     11 3D FF . . .      LXI D,B2DBUF   ;D,E -> FIRST CHAR OF ESC SE
6815     4482     3A 3B FF . . .      LDA B2DEND     ;GET END POINTER
6816     4485     93 . . .      SUB E
6817     4486     3C . . .      INR A
6818     4487     47 . . .      MOV B,A         ;B = NUMBER OF CHARS
6819     4488     CD 67 44 . . .      CALL EXPBF2    ;SEND ESC SEQUENCE
6820     4488     D4 6B 44 . . .      CNC EXPBF3     ;NO ERROR - SEND LAST CHAR
6821     448E     D1 . . .      POP D           ;RESTORE REGISTERS
6822     448F     C1 . . .      POP B
6823     4490     C3 66 44 . . .      JMP EXB050     ;CONTINUE
6824     4493     . . .      STCRLF EQU $   ;STRIP TERM. CR LF, IF ANY
6825     4493     78 . . .      MOV A,B         ;ONLY 1 CHAR LEFT ?
6826     4494     EE 01 . . .      XRI 1Q
6827     4496     C8 . . .      RZ              ;YES-RETURN
6828     4497     7E . . .      MOV A,M         ;GET LAST CHAR
6829     4498     EE 0A . . .      XRI LF         ;IS IT A LF ?
6830     449A     C0 . . .      RNZ             ;NO-DO NOT STRIP
6831     449B     2B . . .      DCX H           ;YES
6832     449C     7E . . .      MOV A,M         ;GET NEXT-TO-LAST CHAR
6833     449D     EE 0D . . .      XRI CR         ;IS IT A CR ?
6834     449F     C0 . . .      RNZ             ;NO DO NOT STRIP
=====

```

13255
2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 201
=====
6835     44A0     05 . .      DCR B          ;YES - DELETE 2 CHARS
6836     44A1     05 . .      DCR B
6837     44A2     C9 . .      RET
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 202
=====
6839     44A3      . . .      ;
6840     44A3      . . .      ; * * * * *
6841     44A3      . . .      ;
6842     44A3      . . .      ;      INTDSP - INITIALIZE DISPLAY FOR COPYING TO
6843     44A3      . . .      ;      BUFFER
6844     44A3      . . .      ;
6845     44A3      . . .      ;      SETS IOFLG2[ENDDSP] IF CURSOR IS BEYOND
6846     44A3      . . .      ;      END OF DISPLAY
6847     44A3      . . .      ;
6848     44A3      . . .      ;
6849     44A3      . . .      ;
6850     44A3      3A 4E FF    INTDS0 EQU $
6851     44A6      FE 04 .     LDA INPDEV ;IS INPUT = DISPLAY?
6852     44A8      C0 . .     CPI DISPLY
6853     44A9      . . .      RNZ ;NO - RETURN
6854     44A9      CD 85 00    INTDSP EQU $
6855     44AC      C8 . .     CALL INITDG ;ANY CHARACTERS?
6856     44AD      . . .      RZ ;YES - RETURN
6857     44AD      21 64 FF    STNDSP EQU $ ;SET END OF DISPLAY FLAG
6858     44B0      7E . .     LXI H,IOFLG2 ;NO - SET END-OF-DISPLAY
6859     44B1      F6 80 .     MOV A,M ;FLAG
6860     44B3      77 . .     ORI ENDDSP
6861     44B4      C9 . .     MOV M,A
RET
=====

```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 204
=====
6913     44DE     C2  F3  44                JNZ  D2B030      ;YES
6914     44E1     .   .   .   ;*****
6915     44E1     CD  CC  47                CALL  CHKFMT     ;FORMAT MODE?
6916     44E4     C2  FB  44                JNZ  D2B040
6917     44E7     3A  C0  FF                LDA  CURROW      ;NO - IN LAST ROW?
6918     44EA     FE  17  .   CPI  MAXROW
6919     44EC     C2  F3  44                JNZ  D2B030      ;NO -
6920     44EF     7E  .   .   MOV  A,M         ;YES - SET DSPBTM
6921     44F0     F6  40  .   ORI  DSPBTM
6922     44F2     77  .   .   MOV  M,A
6923     44F3     .   .   .   D2B030 EQU $
6924     44F3     CD  2A  3D                CALL  GETPTR     ;GET POINTER TO 1ST BYTE
6925     44F6     36  FF  .   MVI  M,-1       ;-1 => NO CHARS REC'D YET
6926     44F8     C3  19  45                JMP  D2B060
6927     44FB     .   .   .   ;*****
6928     44FB     .   .   .   ; WRITE HEADER FOR COPY FROM FORMATTED DISPLAY *
6929     44FB     .   .   .   ;*****
6930     44FB     .   .   .   D2B040 EQU $      ;YES - WRITE HEADER
6931     44FB     CD  2A  3D                CALL  GETPTR     ;GET PTR TO 1ST BYTE OF BUF
6932     44FE     36  C4  .   MVI  M,FLDSEP   ;PUT SEPARATOR
6933     4500     2C  .   .   INR  L
6934     4501     3A  20  FF                LDA  ENDROW      ;CALCULATE NUMBER OF ROWS
6935     4504     47  .   .   MOV  B,A
6936     4505     3A  C0  FF                LDA  CURROW
6937     4508     4F  .   .   MOV  C,A
6938     4509     3A  A3  FF                LDA  TLINO
6939     450C     81  .   .   ADD  C
6940     450D     90  .   .   SUB  B
6941     450E     3D  .   .   DCR  A           ;CURROW + TLINO - ENDROW - 1
6942     450F     77  .   .   MOV  M,A
6943     4510     2C  .   .   INR  L
6944     4511     36  C4  .   MVI  M,FLDSEP   ;PUT SEPARATOR
6945     4513     2C  .   .   INR  L
6946     4514     3A  C1  FF                LDA  CURCOL      ;WRITE CURRENT COLUMN
6947     4517     77  .   .   MOV  M,A
6948     4518     2C  .   .   INR  L
6949     4519     .   .   .   D2B060 EQU $
6950     4519     D5  .   .   PUSH D          ;SAVE PTR TO LENGTH
6951     451A     EB  .   .   XCHG
6952     451B     .   .   .   D2B090 EQU $      ;INSERT CHAR SET FOR FOREIGN
6953     451B     3E  0D  .   MVI  A,STCHST   ;TERMINALS
6954     451D     CD  08  48                CALL  ZKBCTL
6955     4520     D2  25  45                JNC  D2B100
6956     4523     .   .   .   ;*****
6957     4523     .   .   .   ; START FILLING BUFFER FROM DISPLAY *
6958     4523     .   .   .   ; D,E -> BUFFER *
6959     4523     .   .   .   ;*****
6960     4523     .   .   .   D2B099 EQU $
6961     4523     12  .   .   STAX D
6962     4524     1C  .   .   INR  E
=====

```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
6963     4525      . . .      D2B100 EQU $
6964     4525     D5 . . .      PUSH D ;GETDSP KILLS ALL
6965     4526     CD 88 00      CALL GETDSP
6966     4529     D1 . . .      POP D
6967     452A     DA 4B 45      JC D2B200 ;NO CHAR - HANDLE TERMINATOR
6968     452D     FE 0A . . .      CPI LF ;LINE FEED?
6969     452F     C2 23 45      JNZ D2B099 ;NO - PUT IN BUF, GET NXT CH
6970     4532     CD CC 47      CALL CHKFMT ;YES - FORMAT MODE?
6971     4535      . . .      ;*****
6972     4535     3E 0A . . .      MVI A,LF ;RESTORE A
6973     4537      . . .      ;*****
6974     4537     C2 23 45      JNZ D2B099 ;YES - PUT IN BUF, GET NEX
6975     453A     3A C1 FF      LDA CURCOL ;NO - IS LF IN 1ST COLUMN?
6976     453D     3D . . .      DCR A
6977     453E     CA 25 45      JZ D2B100 ;YES - IGNORE LINE FEED
6978     4541     21 64 FF      LXI H,IOFLG2 ;NO - LF IS END OF REC
6979     4544     7E . . .      MOV A,M
6980     4545     E6 BF . . .      ANI -1-DSPBTM ;LF CANNOT BE END OF
6981     4547     77 . . .      MOV M,A ;DISPLAY PAGE
6982     4548     C3 C5 45      JMP D2B440 ;APPEND LF AND QUIT
6983     454B      . . .      ;*****
6984     454B      . . .      ; HANDLE DISPLAY TERMINATORS *
6985     454B      . . .      ;*****
6986     454B      . . .      D2B200 EQU $
6987     454B     FA 6E 45      JM D2B250 ;END OF DISPLAY
6988     454E     C2 8A 45      JNZ D2B300 ;END OF ROW
6989     4551      . . .      ;*****
6990     4551      . . .      ; END OF FIELD *
6991     4551      . . .      ;*****
6992     4551     3A C0 FF      LDA CURROW ;END OF ROW? (HAS ABS ROW NU
6993     4554     21 A3 FF      LXI H,TLINO ;CHANGED?)
6994     4557     86 . . .      ADD M ;CALCULATE ABS ROW NUMBER
6995     4558     21 20 FF      LXI H,ENDROW ;SUBTRACT PREVIOUS ROW NUMBE
6996     455B     96 . . .      SUB M
6997     455C     C2 8A 45      JNZ D2B300 ;YES - BUFFER FINISHED
6998     455F     3E C4 . . .      MVI A,FLDSEP ;NO - PUT FIELD SEPARATOR
6999     4561     12 . . .      STAX D
7000     4562     1C . . .      INR E
7001     4563     3A C1 FF      LDA CURCOL
7002     4566     2E 21 . . .      MVI L,ENDCOL*256/256
7003     4568     96 . . .      SUB M ;CURCOL-ENDCOL
7004     4569     12 . . .      STAX D
7005     456A     1C . . .      INR E
7006     4568     C3 1B 45      JMP D2B090 ;GET MORE CHARACTERS
007     456E      . . .      ;*****
008     456E      . . .      ; END OF DISPLAY *
7009     456E      . . .      ;*****
7010     456E      . . .      D2B250 EQU $
011     456E     2A 73 FF      LHLD GETADR ;WAS LAST CHAR BLOCK TERM?
7012     4571     23 . . .      INX H
=====

```

MEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE 206
7013	4572	3A	04	50	LDA BLKTRM	
7014	4575	BE	.	.	CMP M	
7015	4576	C2	7B	45	JNZ D2B270	;NO -
7016	4579	12	.	.	STAX D	;YES - PUT IT IN THE BUF
7017	457A	1C	.	.	INR E	
7018	457B	.	.	.	D2B270 EQU \$	
7019	457B	CD	08	45	CALL D2B600	;ANY CHARS REC'D?
7020	457E	C2	B3	45	JNZ D2B410	;YES - ADD CR,LF AND RET DAT
7021	4581	.	.	.	D2B280 EQU \$	
7022	4581	E1	.	.	POP H	;NO - RECALL PTR TO LENGTH
7023	4582	36	00	.	MVI M,0	;SET LENGTH = 0
7024	4584	23	.	.	INX H	
7025	4585	36	03	.	MVI M,3	;MARK END OF DISPLAY
7026	4587	C3	CE	45	JMP D2B500	;CLEAR DISP TO BUF FLG & QUI
7027	458A	.	.	.	;*****	
7028	458A	.	.	.	; END OF LINE *	
7029	458A	.	.	.	;*****	
7030	458A	.	.	.	D2B300 EQU \$	
7031	458A	CD	61	60	CALL ZGGTST	;GETTING GRAPHICS DATA?
7032	458D	C2	C1	45	JNZ D2B420	;YES, SKIP OTHER TESTS
7033	4590	.	.	.	;*****	
7034	4590	3A	73	FF	LDA GETADR	;END OF DISPLAY, TOO?
7035	4593	B7	.	.	ORA A	
7036	4594	CC	D8	45	CZ D2B600	;YES - IS THIS A NULL LINE?
7037	4597	CA	81	45	JZ D2B280	;YES - RET END OF DISPLAY
7038	459A	.	.	.	; NO - TREAT AS NORMAL REC,	
7039	459A	.	.	.	; ENDDSP SET BY D2B600	
7040	459A	3A	64	FF	LDA IUFLG2	
7041	459D	E6	40	.	ANI DSPBTM	;BOTTOM OF DISPLAY?
7042	459F	3E	DF	.	MVI A,-1-RECPGE	;IF SO, AND RECORDING FIL
7043	45A1	C4	2A	2B	CNZ CLI0FS	;INHIBIT ROLLUP
7044	45A4	C2	B3	45	JNZ D2B410	;RECORDING PAGE - DO NOT LF
7045	45A7	3A	AE	FF	LDA DSPTYP	;SOFT KEY MODE?
7046	45AA	B7	.	.	ORA A	
7047	45AB	CC	CC	47	CZ CHKFMT	;NO - FORMAT MODE?
7048	45AE	D5	.	.	PUSH D	
7049	45AF	CC	8B	00	CZ LNFEED	;NO - DO LF
7050	45B2	.	.	.	D2B400 EQU \$	
7051	45B2	D1	.	.	POP D	
7052	45B3	.	.	.	D2B410 EQU \$	
7053	45B3	CD	CC	47	CALL CHKFMT	;FORMAT MODE?
7054	45B6	C2	C1	45	JNZ D2B420	;YES - APPEND CR, LF
7055	45B9	1D	.	.	DCR E	;NO - LAST CHAR CR?
7056	45BA	1A	.	.	LDAX D	
7057	45BB	1C	.	.	INR E	
7058	45BC	FE	0D	.	CPI CR	
7059	45BE	CA	C5	45	JZ D2B440	;YES - APPEND LF
7060	45C1	.	.	.	D2B420 EQU \$;APPEND CR, LF
7061	45C1	3E	0D	.	MVI A,CR	
7062	45C3	12	.	.	STAX D	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS
7063	45C4	1C	.	.	INR E
7064	45C5	.	.	D2B440 EQU \$;APPEND LF
7065	45C5	3E	0A	.	MVI A,LF
7066	45C7	12	.	.	STAX D
7067	45C8	1C	.	.	INR E
7068	45C9	E1	.	.	POP H ;RECALL LENGTH PTR
7069	45CA	73	.	.	MOV M,E ;SAVE LENGTH
7070	45CB	23	.	.	INX H
7071	45CC	36	FF	.	MVI M,-1 ;MARK DATA RECORD
7072	45CE	.	.	D2B500 EQU \$	
7073	45CE	23	.	.	INX H ;H,L -> STATUS
7074	45CF	EB	.	.	XCHG ;D,E -> STATUS
7075	45D0	21	64	FF	LXI H,I0FLG2 ;TURN OFF DISP TO BUF FLAG
7076	45D3	7E	.	.	MOV A,M
7077	45D4	E6	DF	.	ANI -1-XDS2BF
7078	45D6	77	.	.	MOV M,A
7079	45D7	C9	.	.	RET
7080	45D8	.	.	.	;*****
7081	45D8	.	.	.	; AT END OF DISP - *
7082	45D8	.	.	.	; NULL REC => RET NZ (END OF DISPLAY) *
7083	45D8	.	.	.	; OTHER => SET ENDDSP & RET Z (DATA RECORD) *
7084	45D8	.	.	.	;*****
7085	45D8	.	.	D2B600 EQU \$	
7086	45D8	62	.	.	MOV H,D ;D,E -> BUFFER, SO SET UP
7087	45D9	2E	00	.	MVI L,0 ;H,L -> FIRST CHAR
7088	45DB	7E	.	.	MOV A,M ;IS 1ST CHAR -1?
7089	45DC	3C	.	.	INR A
7090	45DD	C8	.	.	RZ ;YES => NULL RECORD, RET Z
7091	45DE	C3	AD	44	JMP STNDSP ;NO => SET ENDDSP, RET NZ

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 208
=====
7093     45E1      . . .      ;
7094     45E1      . . .      ; * * * * *
7095     45E1      . . .      ;
7096     45E1      . . .      ;   IOSTGO - TRANSMIT DEVICE STATUS
7097     45E1      . . .      ;
7098     45E1      . . .      ;
7099     45E1      . . .      ;   RETURNS TO SYSTEM VIA "SDTERM"
7100     45E1      . . .      ;
7101     45E1      . . .      IOSTGO EQU $
7102     45E1      3E 1B .      MVI  A,ESC      ;OUTPUT ESCAPE
7103     45E3      CD 7C 00     CALL XPUTD3
7104     45E6      3E 5C .      MVI  A,ABCKSL   ;OUTPUT BACK SLASH
7105     45E8      CD 7C 00     CALL XPUTD3
7106     45EB      3E 70 .      MVI  A,SMALLP   ;OUTPUT SMALL P
7107     45ED      CD 7C 00     CALL XPUTD3
7108     45F0      21 48 FF     LXI  H,IOSTA0   ;GET DEVICE BIT
7109     45F3      7E . .      MOV  A,M
7110     45F4      06 00 .      MVI  B,0
7111     45F6      CD A6 41     CALL BT2NUM      ;CONVERT TO NUMBER
7112     45F9      CD 10 46     CALL IOS120      ;OUTPUT DEVICE
7113     45FC      CD 0E 46     CALL IOS100      ;OUTPUT 1ST STATUS BYTE
7114     45FF      CD 0E 46     CALL IOS100      ;OUTPUT 2ND STATUS BYTE
7115     4602      CD 0E 46     CALL IOS100      ;OUTPUT 3RD STATUS BYTE
7116     4605      . . .      ;
7117     4605      CD 76 00     CALL SDTERM      ;SEND THE TERMINATOR
7118     4608      . . .      ;
7119     4608      . . .      ;   IOSTX1 - CLEAR DEVICE STATUS PENDING
7120     4608      . . .      ;
7121     4608      . . .      IOSTX1 EQU $
7122     4608      01 FF F6     LXI  B,-1-SDVST-SDC2
7123     460B      C3 55 00     JMP  CLBLXF      ;CLEAR FLAG AND EXIT
7124     460E      . . .      ;
7125     460E      . . .      ;   INCREMENT POINTER, GET A BYTE, AND OUTPUT IT
7126     460E      . . .      ;
7127     460E      . . .      IOS100 EQU $
7128     460E      23 . .      INX  H          ;POINT TO NEXT STATUS BYTE
7129     460F      7E . .      MOV  A,M        ;GET THE BYTE
7130     4610      . . .      IOS120 EQU $    ;ENTRY FOR OUTPUTTING DEVICE
7131     4610      F6 30 .      ORI  60Q        ;CONVERT TO ASCII DIGIT
7132     4612      C3 7C 00     JMP  XPUTD3     ;OUTPUT
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 209
7134	4615	.	.	.	;
7135	4615	.	.	.	;
7136	4615	.	.	.	;
7137	4615	.	.	.	;
7138	4615	.	.	.	;
7139	4615	.	.	.	;
7140	4615	.	.	.	;
7141	4615	.	.	.	;
7142	4615	.	.	.	;
7143	4615	.	.	.	;
7144	4615	.	.	.	;
7145	4615	01	FF	7F	IODNGO EQU \$
7146	4618	CD	55	00	LXI B,-1-SDVDUN ;CLEAR DEVICE DONE
7147	4618	3A	4C	FF	CALL CLBLXF ;PENDING FLAG
7148	461E	CD	7C	00	LDA IOCDPT ;FETCH COMPLETION TYPE
7149	4621	C3	76	00	CALL XPUTD3
					JMP SDTERM ;OUTPUT TERMINATOR

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 210
7151	4624	.	.	;	
7152	4624	.	.	; * * * * *	
7153	4624	.	.	;	
7154	4624	.	.	; BF2PRT - PRINT OUT CONTENTS OF BUFFER	
7155	4624	.	.	;	
7156	4624	.	.	; ENTRY: D,E -> BUFFER STATUS	
7157	4624	.	.	;	
7158	4624	.	.	; EXIT : NC => NO ERROR	
7159	4624	.	.	; C => ERROR	
7160	4624	.	.	; D,E -> BUFFER STATUS	
7161	4624	.	.	; A,B,C,H,L DESTROYED	
7162	4624	.	.	;	
7163	4624	.	.	;	
7164	4624	.	.	BF2PRT EQU \$	
7165	4624	CD	41 47	CALL PTRCHK ;PRINTER CONNECTED?	
7166	4627	D8	.	RC ;NO - RETURN	
7167	4628	D5	.	PUSH D ;SAVE PTR TO BUF STATUS	
7168	4629	1B	.	DCX D ;D,E -> TYPE	
7169	462A	1A	.	LDAX D	
7170	462B	3C	.	INR A ;DATA RECORD? (= -1)	
7171	462C	C2	6E 46	JNZ B2P100 ;NO - DO FORM FEED	
7172	462F	3A	FA FF	LDA KBJMP2 ;YES - PASS CONTROL CODES?	
7173	4632	E6	10 .	ANI PRNTAL	
7174	4634	0E	00 .	MVI C,0 ;(SET FOR NOT PASSING)	
7175	4636	CA	3A 46	JZ B2P010	
7176	4639	0D	.	DCR C	
7177	463A	.	.	B2P010 EQU \$	
7178	463A	21	7B 46	LXI H,B2P500 ;PTR TO PRINTOUT ROUTINE	
7179	463D	B4	.	ORA H ;NC,NZ => DO NOT DELETE CRLF	
7180	463E	CD	37 44	CALL EXPBF0 ;EXPAND BUFFER AND PRINT	
7181	4641	.	.	B2P030 EQU \$	
7182	4641	DA	79 46	JC B2P300 ;EXIT ON ERROR	
7183	4644	C2	73 46	JNZ B2P200 ;EXIT ON BUFFER EXHAUSTED	
7184	4647	B7	.	ORA A ;NEXT-TO-LAST CHAR?	
7185	4648	CA	68 46	JZ B2P080 ;YES - OUTPUT LAST	
7186	4648	.	.	;*****	
7187	4648	.	.	; HANDLE FORMAT FIELD SEPARATORS *	
7188	4648	.	.	;*****	
7189	4648	1A	.	LDAX D ;GET FIELD PARAMETER	
7190	464C	B7	.	ORA A ;=0?	
7191	464D	CA	66 46	JZ B2P070 ;YES - IGNORE	
7192	4650	C5	.	PUSH B	
7193	4651	D5	.	PUSH D	
7194	4652	E5	.	PUSH H	
7195	4653	4F	.	MOV C,A ;C <- COUNT	
7196	4654	06	20 .	MVI B,ABLNK ;SET FOR PRINTING ASCII BLAN	
7197	4656	7B	.	MOV A,E ;UNLESS THIS IS THE FIRST	
7198	4657	3D	.	DCR A ;FIELD SEPARATOR	
7199	4658	C2	5D 46	JNZ B2P050	
7200	465B	06	0A .	MVI B,LF ;FIRST FIELD - PRINT LF'S	

=====				PAGE 211	
ITEM	LOC	OBJECT CODE	SOURCE	STATEMENTS	
=====					
7201	465D	. . .	B2P050	EQU \$	
7202	465D	CD E9 46		CALL PTRCHR	;PRINT
7203	4660	E1 . .		POP H	
7204	4661	D1 . .		POP D	
7205	4662	C1 . .		POP B	
7206	4663	DA 79 46		JC B2P300	;EXIT ON ERROR
7207	4666	. . .	B2P070	EQU \$	
7208	4666	13 . .		INX D	;MOVE POINTER PAST COUNT
7209	4667	05 . .		DCR B	
7210	4668	. . .	B2P080	EQU \$	
7211	4668	CD 68 44		CALL EXPBF3	;CONTINUE PRINTING
7212	4668	C3 41 46		JMP B2P030	;EVALUATE RESULTS
7213	466E	. . .		;*****	
7214	466E	. . .		; PRINT FORM FEED *	
7215	466E	. . .		;*****	
7216	466E	. . .	B2P100	EQU \$	
7217	466E	3E 0C .		MVI A,FF	
7218	4670	CD 7B 46		CALL B2P500	
7219	4673	. . .		;*****	
7220	4673	. . .		; RELEASE BUFFER AND QUIT *	
7221	4673	. . .		;*****	
7222	4673	. . .	B2P200	EQU \$	
7223	4673	D1 . .		POP D	;D,E -> STATUS
7224	4674	1A . .		LDAX D	
7225	4675	E6 F7 .		ANI -1-PRINTR	
7226	4677	12 . .		STAX D	
7227	4678	C9 . .		RET	;RET NC => NO ERROR
7228	4679	. . .		;*****	
7229	4679	. . .		; ERROR RETURN *	
7230	4679	. . .		;*****	
7231	4679	. . .	B2P300	EQU \$	
7232	4679	D1 . .		POP D	;D,E -> STATUS
7233	467A	C9 . .		RET	;IOEXIT CLEARS BUFS
7234	467B	. . .		;*****	
7235	467B	. . .		; ROUTINE TO OUTPUT ONE BYTE *	
7236	467B	. . .		;*****	
7237	467B	. . .	B2P500	EQU \$	
7238	467B	C5 . .		PUSH B	
7239	467C	0E 01 .		MVI C,1	;ONE REPETITION
7240	467E	47 . .		MOV B,A	;BYTE IN B-REG
7241	467F	D5 . .		PUSH D	
7242	4680	CD E9 46		CALL PTRCHR	
7243	4683	D1 . .		POP D	
7244	4684	C1 . .		POP B	
7245	4685	C9 . .		RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 212
7247	4686	.	.	. ;*****	
7248	4686	.	.	. ; 9866 PRINTER DRIVER *	
7249	4686	.	.	. ;*****	
7250	4686	.	.	. ;	
7251	4686	.	.	. ;	
7252	4686	.	.	. ;	
7253	4686	.	.	. PRCHR1 EQU \$	
7254	4686	21	DC 05	LXI H,PTDLY ;SET TIMEOUT COUNTER	
7255	4689	.	.	. PTR110 EQU \$	
7256	4689	3A	00 8D	LDA PTRST1 ;INPUT STATUS	
7257	468C	B7	.	. ORA A ;IS PRINTER OUT OF PAPER?	
7258	468D	F2	49 47	JP PTR700 ;YES-TERMINATE	
7259	4690	0F	.	. RRC ;IS PRINTER READY?	
7260	4691	57	.	. MOV D,A ;SAVE STATUS	
7261	4692	DA	A1 46	JC PTR120 ;NO - CHECK TIMEOUT	
7262	4695	78	.	. MOV A,B ;GET CHARACTER	
7263	4696	32	20 8D	STA PTRST1 ;OUTPUT	
7264	4699	7A	.	. MOV A,D ;GET STATUS	
7265	469A	E6	02 .	ANI 2 ;STROBE DATA?	
7266	469C	C0	.	. RNZ ;NO - QUIT	
7267	469D	3A	02 8D	LDA PTRCL1 ;YES - OUTPUT STROBE	
7268	46A0	C9	.	. RET ;AND RETURN	
7269	46A1	.	.	. PTR120 EQU \$	
7270	46A1	3A	F6 FF	LDA INTFLG ;TIMER INTERRUPT?	
7271	46A4	D6	03 .	SUI TMRINT	
7272	46A6	C2	89 46	JNZ PTR110 ;NO - CONTINUE WAITING	
7273	46A9	32	F6 FF	STA INTFLG ;YES - CLEAR FLAG	
7274	46AC	2B	.	. DCX H ;DECREMENT TIMEOUT COUNTER	
7275	46AD	7C	.	. MOV A,H ;TIME OUT?	
7276	46AE	B5	.	. ORA L	
7277	46AF	CA	49 47	JZ PTR700 ;YES-REPORT ERROR	
7278	46B2	C3	89 46	JMP PTR110 ;CHECK STATUS AGAIN	

				PAGE 213	
ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	
7280	46B5	.	.	.	;*****
7281	46B5	.	.	.	; RS-232 PRINTER DRIVER *
7282	46B5	.	.	.	;*****
7283	46B5	.	.	.	;
7284	46B5	.	.	.	PRCHR2 EQU \$
7285	46B5	21	DC	05	LXI H,PTDLY ;SET TIMEOUT COUNTER
7286	46B8	.	.	.	PTR630 EQU \$
7287	46B8	3A	20	85	LDA PTRST2 ;INPUT STATUS
7288	46BB	.	.	.	;*****
7289	46BB	.	.	.	; MONITOR XMIT REG EMPTY ONLY, NOT CLEAR TO SEND
7290	46BB	E6	02	.	ANI PTRDY2 ;PRINTER READY?
7291	46BD	CA	D5	46	JZ PTR640 ;NO, WAIT
7292	46C0	.	.	.	;*****
7293	46C0	3A	40	85	LDA PTRCF2 ;READ IN CONFIG. STRAPS
7294	46C3	E6	E0	.	ANI PTRHD2 ;IS IT A HANDSHAKE DEVICE?
7295	46C5	C2	D0	46	JNZ PTR635 ;NO - OUTPUT THE CHARACTER
7296	46C8	3A	20	85	LDA PTRST2 ;GET STATUS
7297	46C8	E6	40	.	ANI PTRSB2 ;IS SB LINE SET/RDY?
7298	46CD	CA	D5	46	JZ PTR640 ;NO-GO WAIT FOR PRINTER
7299	46D0	.	.	.	PTR635 EQU \$
7300	46D0	.	.	.	;
7301	46D0	78	.	.	MOV A,B ;GET CHARACTER
7302	46D1	32	60	85	STA PTRDA2 ;OUTPUT
7303	46D4	C9	.	.	RET
7304	46D5	.	.	.	PTR640 EQU \$
7305	46D5	3A	F6	FF	LDA INTFLG ;TIMER INTERRUPT
7306	46D8	D6	03	.	SUI TMRINT
7307	46DA	C2	B8	46	JNZ PTR630 ;NO - CONTINUE WAITING
7308	46DD	32	F6	FF	STA INTFLG ;YES - CLEAR FLAG
7309	46E0	2B	.	.	DCX H ;DECREMENT TIMEOUT COUNTER
7310	46E1	7C	.	.	MOV A,H ;TIME OUT?
7311	46E2	B5	.	.	ORA L
7312	46E3	CA	49	47	JZ PTR700 ;YES-REPORT ERROR
7313	46E6	C3	B8	46	JMP PTR630 ;NO-GO CHECK STATUS

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 215
7365	4709	D8	.	RC ;ERROR EXIT	
7366	470A	.	.	*****	
7367	470A	.	.	; IF ALL SWITCHES CLOSED, DONT SEND ANY NULLS	
7368	470A	3A	40 85	LDA PTRCF2 ;FETCH CONFIGURATION STRAPS	
7369	470D	E6	E0 .	ANI PTRHD2 ;GET NO. OF NULLS STRAPS	
7370	470F	6F	.	MOV L,A ;SAVE IN L	
7371	4710	FE	E0 .	CPI PTRHD2 ;ALL SWITCHES CLOSED?	
7372	4712	CA	3C 47	JZ PTRC90 ;YES, DONT SEND ANY NULLS	
7373	4715	.	.	*****	
7374	4715	1E	01 .	MVI E,1 ;SET UP 1 NULL OUTPUT	
7375	4717	78	.	MOV A,B ;GET CHARACTER	
7376	4718	FE	20 .	CPI 40Q ;IS IT GREATER THAN 37B?	
7377	471A	D2	3C 47	JNC PTRC90 ;YES - EXIT	
7378	471D	FE	0D .	CPI CR ;NO-IS IT A CARRIAGE RET?	
7379	471F	C2	28 47	JNZ PTRC70 ;NO - GO ADD FILL PAD	
7380	4722	7A	.	MOV A,D ;RECALL NEXT CHAR	
7381	4723	FE	0A .	CPI LF ;IS NEXT CHARACTER A LF?	
7382	4725	CA	30 47	JZ PTRC80 ;YES - OUTPUT ONE NULL	
7383	4728	.	.	PTRC70 EQU \$;NO - OUTPUT NULLS	
7384	4728	.	.	*****	
7385	4728	7D	.	MOV A,L ;RECALL NO. OF NULLS	
7386	4729	B7	.	ORA A ;IS PRINTER HANDSHAKE DEVICE	
7387	472A	.	.	*****	
7388	472A	CA	30 47	JZ PTRC80 ;NO - OUTPUT ONE NULL	
7389	472D	1F	.	RAR ;GET THE NUMBER OF FILLS	
7390	472E	1F	.	RAR	
7391	472F	5F	.	MOV E,A ;SET UP NULL COUNTER	
7392	4730	.	.	PTRC80 EQU \$	
7393	4730	06	00 .	MVI B,0 ;SETUP NULL CHARACTER	
7394	4732	.	.	PTRC85 EQU \$	
7395	4732	CD	B5 46	CALL PRCHR2 ;OUTPUT THE CHARACTER	
7396	4735	D8	.	RC ;ERROR EXIT	
7397	4736	1D	.	DCR E ;IS THIS THE LAST NULL?	
7398	4737	C2	32 47	JNZ PTRC85 ;NO - GO DO IT AGAIN	
7399	473A	06	0A .	MVI B,LF ;RESTORE B TO A LF	
7400	473C	.	.	PTRC90 EQU \$;YES - COMPLETE	
7401	473C	.	.	;	
7402	473C	0D	.	DCR C ;IS THIS THE LAST ONE?	
7403	473D	C2	06 47	JNZ PTRC60 ;NO - DO NEXT	
7404	4740	C9	.	RET ;EXIT	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 216
=====
7406     4741      . . .      ;
7407     4741      . . .      ; PTRCHK-CHECKS FOR PRINTER ON-LINE
7408     4741      . . .      ;
7409     4741      . . .      ; ENTRY: A,B,C,D,E,L=DON'T CARE
7410     4741      . . .      ;
7411     4741      . . .      ; EXIT : C => ERROR, NO PRINTER
7412     4741      . . .      ; NC => NO ERROR
7413     4741      . . .      ; A,H,L DESTROYED
7414     4741      . . .      ;
7415     4741      . . .      ;
7416     4741      . . .      PTRCHK EQU $
7417     4741      21 78 FE LXI H, PTRABT ;GET PRINTER ERROR FLAG
7418     4744      AF . . XRA A ;CLEAR A REGISTER
7419     4745      77 . . MOV M,A ;CLEAR PRINTER ERROR FLAG
7420     4746      2D . . DCR L ;GET PRINTER FLAG
7421     4747      B6 . . ORA M ;ANY PRINTERS CONNECTED?
7422     4748      C0 . . RNZ ;YES - EXIT WITH NC
7423     4749      . . .      ;
7424     4749      . . .      ;
7425     4749      . . .      ; DRIVER ERROR RETURN
7426     4749      . . .      ;
7427     4749      . . .      PTR700 EQU $
7428     4749      21 AB 3B LXI H,PREMSG ;SET PRINTER ERROR MESSAGE
7429     474C      CD ED 2D CALL CTUER1 ;REPORT ERROR (SETS C-BIT)
7430     474F      21 78 FE LXI H, PTRABT ;SET PRINTER ERROR FLAG
7431     4752      36 FF . MVI M,-1
7432     4754      C9 . . RET ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
7434	4755	.	.	;
7435	4755	.	.	;
7436	4755	.	.	;
7437	4755	.	.	;
7438	4755	.	.	;
7439	4755	.	.	;
7440	4755	.	.	;
7441	4755	.	.	;
7442	4755	.	.	;
7443	4755	.	.	;
7444	4755	.	.	;
7445	4755	.	.	;
7446	4755	.	.	;
7447	4755	.	.	STPRT EQU \$
7448	4755	AF	.	XRA A ;CLEAR A REGISTER
7449	4756	21	49 FF	LXI H,IOSTA1 ;CLEAR IOSTA1
7450	4759	77	.	MOV M,A
7451	475A	2C	.	INR L ;SKIP OVER STAT 2
7452	475B	2C	.	INR L ;CLEAR STAT3
7453	475C	77	.	MOV M,A
7454	475D	3A	78 FE	LDA PTRABT ;GET PRINTER ERROR FLAG
7455	4760	E6	08 .	ANI 100 ;MASK FOR ERROR STATUS
7456	4762	EE	08 .	XRI 100 ;SET PROPER VALUE
7457	4764	32	4A FF	STA IOSTA2 ;SET IOSTA2 VALUE
7458	4767	3A	77 FE	LDA PTRFLG ;GET PRINTER FLAG
7459	476A	1F	.	RAR ;IS IT DRIVER 2?
7460	476B	D2	8B 47	JNC PTRS80 ;YES - BUILD RS-232 STATUS
7461	476E	.	.	;
7462	476E	.	.	;
7463	476E	.	.	;
7464	476E	3A	00 8D	LDA PTRST1 ;GET PRINTER STATUS
7465	4771	B7	.	ORA A ;IS PRINTER ON-LINE?
7466	4772	CA	84 47	JZ PTRS10 ;NO-GO REPORT NO PRINTER
7467	4775	2E	4B .	MVI L,IOSTA3 ;GET IOSTA3 POINTER
7468	4777	36	01 .	MVI M,1 ;SET ON-LINE BIT
7469	4779	07	.	RLC ;PRINTER OUT OF PAPER?
7470	477A	DA	81 47	JC PTRS20 ;NO - CHECK READY STATUS
7471	477D	2E	49 .	MVI L,IOSTA1 ;INSERT PAPER OUT
7472	477F	36	02 .	MVI M,2 ;SET PAPER OUT STATUS
7473	4781	.	.	PTRS20 EQU \$
7474	4781	0F	.	RRC
7475	4782	0F	.	RRC ;PRINTER READY?
7476	4783	D0	.	RNC ;YES - RETURN
7477	4784	.	.	PTRS10 EQU \$;NO-SETUP PRINTER BUSY
7478	4784	3E	01 .	MVI A,1 ;INSERT PRINTER BUSY
7479	4786	2E	4A .	MVI L,IOSTA2 ;IN IOSTA2
7480	4788	B6	.	ORA M ;OR IN COMMAND REJECT
7481	4789	77	.	MOV M,A ;AND SAVE STATUS
7482	478A	C9	.	RET
7483	478B	.	.	;

13255

2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 218
=====
7484     478B      .   .   .   ;      RS-232 STATUS
7485     478B      .   .   .   ;
7486     478B      .   .   .   PTRS80 EQU $
7487     478B      3A  40  85   LDA PTRCF2 ;GET CONFIG. STRAPS
7488     478E      4F   .   .   MOV C,A ;SAVE STRAP STATUS IN C
7489     478F      E6  1F   .   ANI PTRBD2 ;ISOLATE BAUD AND PARITY
7490     4791      17   .   .   RAL ;SETUP BAUD RATE FOR OUTPUT
7491     4792      2E  4B   .   MVI L,IOSTA3 ;SAVE IT IN IOSTA3
7492     4794      77   .   .   MOV M,A
7493     4795      3A  20  85   LDA PTRST2 ;GET PRINTER STATUS
7494     4798      47   .   .   MOV B,A ;SAVE PRINTER STATUS IN B
7495     4799      E6  22   .   ANI PTRDY2+PTROL2 ;IS PRINTER READY?
7496     479B      FE  02   .   CPI PTRDY2
7497     479D      C2  84  47   JNZ PTRS10 ;NO-REPORT PRINTER BUSY
7498     47A0      3E  01   .   MVI A,1 ;SETUP PRINTER ON-LINE
7499     47A2      B6   .   .   ORA M ;OR IN BAUD RATE
7500     47A3      77   .   .   MOV M,A ;SET ON-LINE BIT
7501     47A4      79   .   .   MOV A,C ;GET CONFIGURATION STRAPS
7502     47A5      E6  E0   .   ANI PTRHD2 ;IS IT A HANDSHAKE DEVICE?
7503     47A7      C0   .   .   RNZ ;NO-EXIT
7504     47A8      78   .   .   MOV A,B ;GET STATUS
7505     47A9      E6  40   .   ANI PTRSB2 ;IS SB LINE SET/ROY?
7506     47AB      CA  84  47   JZ PTRS10 ;NO-GO REPORT BUSY
7507     47AE      C9   .   .   RET
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
7509	47AF	.	.	;
7510	47AF	.	.	;
7511	47AF	.	.	;
7512	47AF	.	.	;
7513	47AF	.	.	;
7514	47AF	.	.	;
7515	47AF	.	.	;
7516	47AF	.	.	;
7517	47AF	.	.	;
7518	47AF	.	.	;
7519	47AF	.	.	;
7520	47AF	.	.	;
7521	47AF	.	.	;
7522	47AF	.	.	;
7523	47AF	.	.	;
7524	47AF	.	.	;
7525	47AF	.	.	;
7526	47AF	.	.	;
7527	47AF	.	.	;
7528	47AF	.	.	;
7529	47AF	.	.	;
7530	47AF	CD	41	47
7531	47B2	D8	.	.
7532	47B3	21	D5	FF
7533	47B6	4E	.	.
7534	47B7	06	0A	.
7535	47B9	2E	D8	.
7536	47BB	56	.	.
7537	47BC	15	.	.
7538	47BD	CA	E9	46
7539	47C0	.	.	.
7540	47C0	0E	01	.
7541	47C2	06	0C	.
7542	47C4	3E	19	.
7543	47C6	32	20	87
7544	47C9	C3	E9	46

CTLPRT-- OUTPUT STANDARD CONTROLS TO
PRINTER

WHERE C IS DEFINED AS FOLLOWS:

- 0 - NOT DEFINED (FF)
- 1 - LINE FEED (P=# OF LF'S)
- 2 - TOP OF FORM (FF)
- 3 - NOT DEFINED (FF)
- 4 - NOT DEFINED (FF)
- 5 - END OF FILE (FF)
- 6 - END OF VALID (FF)
DATA

EXIT: A,B,C,D,E,L DESTROYED
H= BASEH

```

CTLPRT EQU $
CALL PTRCHK ;IS A PRINTER ON-LINE
RC ;NO - EXIT
LXI H,IOCCNT ;SET UP NUM. OF LINE FEEDS
MOV C,M
MVI B,LF ;GET ASCII LF
MVI L,IOCTYP ;GET TYPE OF CONTROL
MOV D,M
DCR D ;IS IT A LF REQUEST?
JZ PTRCHR ;YES-GO DO LINE FEEDS
PRTX05 EQU $
MVI C,1 ;SETUP FOR ONE CHARACTER
MVI B,FF ;SETUP FOR FORM FEED=FF
MVI A,25 ;MOVE CURSOR OFF SCREEN
STA IOCRRW
JMP PTRCHR ;GO OUTPUT CHARACTER
    
```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 220
=====
7546     47CC      . . .      ;
7547     47CC      . . .      ; EQUATES FROM MAIN CODE SECTION
7548     47CC      . . .      ; TO BE REMOVED WHEN CODE IS MERGED
7549     47CC      . . .      ;*****
7550     47CC      . . .      ;
7551     0098      . . .      ENTRCD EQU 230Q      ;ENTER KEY CODE
7552     009E      . . .      SLKYCD EQU 236Q      ;SELECT KEY CODE
7553     00A0      . . .      FNCLIM EQU 240Q      ;FUNCTION CODE UPPER LIMIT
7554     00FA      . . .      TESTKY EQU 372Q      ;TEST KEY CODE
=====
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
7556	47CC	.	.	;
7557	47CC	.	.	; COMMON ROUTINES - DUPLICATED FROM MAIN CODE
7558	47CC	.	.	;
7559	47CC	.	.	;
7560	4FFD	.	.	ZZZZZ EQU 47775Q ;SLEEP ROUTINE
7561	47CC	.	.	;
7562	47CC	.	.	;
7563	47CC	.	.	; * * * * *
7564	47CC	.	.	;
7565	47CC	.	.	; CHKFMT - CHECK FORMAT MODE
7566	47CC	.	.	;
7567	47CC	.	.	; ENTRY: H = BASEH
7568	47CC	.	.	;
7569	47CC	.	.	; EXIT : Z = TRUE, NOT FORMAT MODE
7570	47CC	.	.	; Z = FALSE, FORMAT MODE
7571	47CC	.	.	; A,L DESTROYED
7572	47CC	.	.	;
7573	47CC	.	.	CHKFMT EQU \$
7574	47CC	3A	F4 FF	LDA MDFLG1 ;GET TERMINAL MODE FLAGS
7575	47CF	E6	08 .	ANI FORMAT ;MASK FOR FORMAT FLAG
7576	47D1	C9	.	RET ;RETURN
7577	47D2	.	.	;
7578	47D2	.	.	;*****
7579	47D2	.	.	; SEND CR(LF) TO DATACOM *
7580	47D2	.	.	;*****
7581	47D2	.	.	SDCRLF EQU \$
7582	47D2	3E	0D .	MVI A,CR ;SEND THE CR
7583	47D4	CD	A7 42	CALL IOR400
7584	47D7	.	.	;*****
7585	47D7	.	.	; SEND LF IFF AUTO LF DEPRESSED *
7586	47D7	.	.	;*****
7587	47D7	.	.	SDAULF EQU \$
7588	47D7	3A	F3 FF	LDA MDFLG2 ;AUTO LF DEPRESSED?
7589	47DA	E6	04 .	ANI AUTOLF
7590	47DC	C8	.	RZ ;NO - RETURN
7591	47DD	3E	0A .	MVI A,LF ;YES - SEND LF
7592	47DF	C3	A7 42	JMP IOR400
7593	47E2	.	.	;*****
7594	47E2	.	.	DCERR1 EQU \$
7595	47E2	C2	9D 00	JNZ HANGUO
7596	47E5	3A	F4 FF	LDA MDFLG1
7597	47E8	E6	40 .	ANI RECORD
7598	47EA	CA	B3 41	JZ IOFAIL
7599	47ED	C3	34 48	JMP USRINT
7600	47F0	.	.	;*****
7601	47F0	.	.	ERRCHK EQU \$
7602	47F0	3A	4F FF	LDA IOCERR
7603	47F3	FE	46 .	CPI F
7604	47F5	CA	45 43	JZ SDEOF
7605	47F8	C3	75 43	JMP RDABRT

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 222
=====
7607     47FB      . . .      ;
7608     47FB      . . .      ; VECTORS INTO MAIN CODE
7609     47FB      . . .      ;
7610     0040      . . .      MANORG EQU 1000
7611     0040      . . .      DSPMSG EQU MANORG ;DISPLAY A MESSAGE
7612     0043      . . .      RSTDSP EQU DSPMSG+3 ;RESTORE NORMAL DISPLAY
7613     0046      . . .      DCNUM EQU RSTDSP+3
7614     0049      . . .      DCPLUS EQU DCNUM+3
7615     004C      . . .      DCMNUS EQU DCPLUS+3
7616     004F      . . .      ESCEND EQU DCMNUS+3
7617     0052      . . .      CHKLIM EQU ESCEND+3
7618     0055      . . .      CLBLXF EQU CHKLIM+3
7619     0058      . . .      SBLXF0 EQU CLBLXF+3
7620     005B      . . .      SBLXFA EQU SBLXF0+3
7621     005E      . . .      STRTBL EQU SBLXFA+3
7622     0061      . . .      CURPH EQU STRTBL+3
7623     0064      . . .      CURPHD EQU CURPH+3
7624     0067      . . .      FRECNT EQU CURPHD+3
7625     006A      . . .      PTBLK0 EQU FRECNT+3
7626     006D      . . .      CLEARL EQU PTBLK0+3
7627     0070      . . .      CLEARS EQU CLEARL+3
7628     0073      . . .      FNDBT2 EQU CLEARS+3
7629     0076      . . .      SDTERM EQU FNDBT2+3
7630     0079      . . .      XDTRM1 EQU SDTERM+3 ;OUTPUT TERMINATOR ONLY
7631     007C      . . .      XPUTD3 EQU XDTRM1+3 ;TRANSMIT CHARACTER
7632     007F      . . .      TEST EQU XPUTD3+3
7633     0082      . . .      CHINT EQU TEST+3
7634     0085      . . .      INITDG EQU CHINT+3
7635     0088      . . .      GETDSP EQU INITDG+3
7636     008B      . . .      LNFEED EQU GETDSP+3
7637     008E      . . .      EXPAND EQU LNFEED+3
7638     0091      . . .      NXTCHR EQU EXPAND+3
7639     0094      . . .      GETDCM EQU NXTCHR+3
7640     0097      . . .      MLKSCH EQU GETDCM+3
7641     009A      . . .      MLKOFF EQU MLKSCH+3
7642     009D      . . .      HANGU0 EQU MLKOFF+3
7643     00A0      . . .      BUFMSG EQU HANGU0+3
7644     00A2      . . .      DCTEST EQU BUFMSG+2
7645     00A5      . . .      IORMG0 EQU DCTEST+3
7646     00A8      . . .      BN2DEC EQU IORMG0+3 ;CONVERT BINARY TO DECIMAL
7647     00AB      . . .      BN2DE0 EQU BN2DEC+3 ;CONVERT 1 BYTE BINARY TO DE
7648     00AE      . . .      RCADRA EQU BN2DE0+3 ;LOCATE CURRENT CURSOR POS
7649     00B1      . . .      GTMODE EQU RCADRA+3 ;CHECK FOR PAGE MODE
=====
  
```

13255
2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

=====

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 223
------	-----	-------------	-------------------	----------

=====

7651	47FB	.	.	.	END
0	ERRORS FOUND IN ASSEMBLY CODE .				

13255
 2648A MICROCODE LISTING 'I0273'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
A      0041      344
ABCKSL 005C      358, 7104
ABLNK  0020      326, 7196
ABSTAK FF5F      767, 769, 955, 1518, 1779, 5730, 5731, 5751
ADEL   007F      373
ALCC   0063      364
ALPHA  00C5      384
ALPHNM 00C7      386
ALT100 412F      5948, 5939, 5942, 5945
ALT2BF 412D      5946, 5974
ALT500 43B4      6570, 5953
ALTIN  0040      873
ALTIO  0010      790
ALTRG  9200      301, 303, 5949
ALTOU  482A      230, 6751
AMPSND 0026      327
ANL    0080      782, 1409, 1463, 1476, 1481, 1489
ANR    0040      781, 1412, 1463, 1476, 1481, 1491
ARPARN 0029      329
ATSIGN 0040      343
AUTOLF 0004      149, 7589
AUTTRM 0001      80
B15    8000      394, 5082, 5085, 5087, 5090, 5103, 5105, 5110
B1LEN  FF38      835, 836, 3331, 3370, 5050, 5070, 6440
B1STAT FF3A      833, 834, 1305, 4553, 5014, 5033, 5051, 6184, 6545
B1TYPE FF39      834, 835, 3301, 6199
B2C015 3013      2755, 2748
B2C030 3028      2773, 2776, 2827
B2C040 302F      2777
B2C050 3041      2786, 2788
B2C100 305D      2806, 2798
B2C105 3066      2812, 2550
B2C110 306C      2818, 2808
B2C150 3080      2828, 2824
B2D003 43E0      6622, 6620
B2D005 43E6      6626, 6652
B2D010 43F8      6641, 6628
B2D020 4405      6650, 6636
B2D030 440E      6657, 6614, 6625, 6651
B2D100 4419      6669, 6616
B2D120 441C      6672, 6621, 6692, 6694
B2D200 4425      6683, 6647, 6656
B2DBFL 003D      817
B2DBUF FF3D      816, 817, 818, 3344, 4390, 6814
B2DEND FF3B      819, 833, 6815
B2DPTR FF3C      818, 819
B2LEN  FF35      838, 842, 3332, 6442
B2P010 463A      7177, 7175
B2P030 4641      7181, 7212
B2P050 465D      7201, 7199
B2P070 4666      7207, 7191
B2P080 4668      7210, 7185
B2P100 466E      7216, 7171
  
```

13255
2648A MICROCODE LISTING 'I0273'

SYMBOL	VALUE	REFERENCED ON
B2P200	4673	7222, 7183
B2P300	4679	7231, 7182, 7206
B2P500	467B	7237, 7178, 7218
B2STAT	FF37	836, 837, 1308, 4538, 5018, 5034, 5053, 6186, 6550
B2TYPE	FF36	837, 838, 3302, 6201
BAKSPR	2C79	2012, 1176, 2709, 2779
BAKSPW	2C74	2009, 2146, 6555
BAKSPX	2C81	2017, 3544
BASE	FF00	531, 817, 1421, 1936, 2390, 2958
BASE2	FE00	533
BASEH	00FF	530, 531, 532, 1140, 5297
BASEH2	00FE	532, 533
BELLIM	0008	403
BF2ALT	4128	5943, 5965
BF2DSP	43C4	6605, 5963
BF2DTB	4139	5960, 6048
BF2LCT	2FF0	2731, 5961
BF2PRT	4624	7164, 5964
BF2RCT	2FF5	2734, 5962
BFSPCE	CFFF	603
BINXMT	0002	407, 5784, 6207
BLKFIL	FF91	600, 601
BLKMDE	0002	148, 4163, 4175
BLKMSG	3C61	4826, 4381
BLKSM	000F	404
BLKSZ	0010	405
BLKTRG	0001	104
BLKTRM	5004	258, 259, 6311, 6498, 7013
BN2DE0	00AB	7647, 7648, 4395, 4858, 5737
BN2DEC	00A8	7646, 7647, 4396, 5765
BNABRT	4372	6529, 6446
BNR005	431B	6443, 6441
BNR010	432A	6452, 4925, 6445
BNR020	4336	6461, 6467
BNRYGO	430D	6436, 910, 6419
BOT	0020	751, 1064, 1069, 1095, 1257, 1711, 1736, 1758, 1774
BSP010	2C7B	2014, 2011
BSP015	2C8B	2024, 2021
BSP020	2C8D	2026, 2015
BSY010	2DCD	2308, 2314
BSYCHK	2DC7	2305, 916, 2316, 3511, 3521, 5509, 5688
BSYCK0	2DC1	2301, 3205, 4191, 4907
BSYMSG	3BF8	4810, 2309
BT2NUM	41A6	6110, 6079, 6113, 7111
BUF2CT	2FF7	2736, 2166, 2206, 2733, 2751, 2990, 3884
BUFBGN	FF8D	605, 606
BUFBSY	0080	793, 2800, 2895, 3003, 3161
BUFEND	FF8B	606, 610
BUFMSG	00A0	7643, 7644, 4637
C	0043	345
C2B020	2E30	2384, 2375, 2379
CALTST	FF75	636, 637, 6752
CAPSLK	0001	147

13255

2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
CARDIO 3CA3 4870, 1127, 1816, 2310, 2706
CARDIS 3CA6 4888, 3561, 3752, 3975, 4332
CDSPEN FF77 634, 635
CHAR FF88 612, 613, 4270, 4374, 5325, 5902
CHARIN FF9C 591, 592, 4076, 4085
CHEKCC 0040 97
CHGBUF 3D1F 5048, 2372, 2376, 3166
CHINT 0082 7633, 7634, 6589
CHKCMD 3EAA 5392, 5213, 5234, 5249, 5306
CHKEOF 2AFC 1586, 2022, 2124, 2915, 2974, 3597, 3634
CHKEOV 2B0E 1617, 2003, 2143, 2928, 2971, 3211, 3837, 3843
CHKESD 2B18 1633, 2421, 3649
CHKEW 2B02 1600, 2005, 2918, 3213
CHKFMT 47CC 7573, 4255, 5526, 5575, 6345, 6684, 6915, 6970, 7047, 7053
CHKFWD 2AE7 1555, 1037, 1045, 1066, 1097, 1943, 2222, 3693, 3773
CHKLIM 0052 7617, 7618, 3609
CHKLPM 2BAD 1808, 1177, 2001, 2147, 2434, 2710, 2863, 3648, 3745
CHKRTN FF86 613, 614
CHRSET FF72 641, 642
CHSAV FF98 595, 597
CIL 0001 722, 934, 1229, 1247, 1280, 1387
CIR 0002 721, 934, 1229, 1249, 1283, 1387
CISCAN 2AED 1569, 1294, 1690, 1812, 2420, 2857, 3535, 3623
CKBRKY 000A 243
CKD100 3935 4425, 4417, 4419, 4421
CKDVKY 391F 4414, 4082, 4343, 4450
CKIOKY 0008 241
CLBLXF 0055 7618, 7619, 5369, 6164, 6455, 6562, 7123, 7146
CLEARL 006D 7626, 7627, 4697
CLEARS 0070 7627, 7628, 4258
CLIOFS 2B2A 1667, 1219, 2791, 4475, 4910, 5348, 5521, 5560, 5901, 6509,
6511, 7043
CLRCTO 32A5 3234, 1181, 3534, 3661
CLRTRG 0000 282
CLRTRM 0002 81
CLWBSR 36DC 4030, 3506, 4013
CLXB2D 43AC 6563, 6315, 6663
CMBASE 00FF 165, 166
CMDEXC 0008 749, 1469, 1702, 1774, 2327, 3457, 3515, 3841
CMFLGS FFF8 174, 175
CMND FF55 773, 795, 1007, 1159, 1286, 1297, 1381, 1421, 1479, 1507,
1541, 1556, 1877, 1894, 1936, 2112, 2263, 2370, 2390, 2852,
3404
CMP100 3F84 5608, 5605
CMPBFS 32C3 3300, 5615
CMPFCN 3F7C 5602, 5496
CMPLIM FF46 815, 816, 3346, 5474, 5524, 5552, 5589
CMSTOR FF00 166
CNTFAD FFCE 538
CNTRL0 FF62 755, 766, 1258, 1587, 1618, 1634, 1699, 1923, 2178, 2423,
2463, 2545, 2669, 2756, 2794, 2822, 2950, 3017, 3185, 3235,
3738, 3775, 3886, 4712, 4714
CNTXFR 0002 669

```

2648A MICROCODE LISTING 'I0273'

SYMBOL VALUE REFERENCED ON

```

=====
COMAMS 3C93 4844, 5692
COMMA 002C 331
COMMON FFFF 164, 165, 168
CONDIS 0001 51
CONDTN 364C 3906, 903
CONMSG 3C78 4832, 6013
COUNT FF84 618, 619
CPB010 32D2 3308, 3306
CPB040 32DD 3314, 3312
CPB100 32F3 3327, 3305
CPB300 32F5 3330
CPB310 3308 3339, 3391
CPB320 3333 3357, 3349
CPB330 3340 3363, 3348
CPB350 3342 3366, 3334
CPB360 334D 3372, 3379
CPB370 335C 3383, 3377
CR 000D 319, 4595, 6301, 6691, 6833, 7058, 7061, 7378, 7582
CRAFLG FF67 691, 713
CRTOFF 0080 435
CT2BUF 2E12 2369, 1186, 2366, 2382, 3705
CTBDLY 0020 798, 1500
CTBLNK FF53 796, 797, 1416, 1503
CTBLTM FF52 797, 799, 1501
CTCTLT 33D2 3495, 3518
CTD050 3CC6 4913, 4927
CTD100 3CE3 4934, 4916
CTD110 3CE4 4936, 4922
CTDCDP 3CAF 4903, 911
CTH100 2908 1133, 1138
CTHANG 28F4 1122, 1119, 1296
CTHNG0 28EB 1117, 1091
CTHNG1 28F1 1120, 1039
CTI020 2866 953, 951
CTI040 2871 960, 957
CTI100 2873 962, 948
CTIADR FF33 842, 843, 1697, 1714, 2487, 2532, 2555, 2629, 2963, 3011,
3028, 3073, 3129, 3787, 3791, 3818
CTIBPT FF2F 844, 845, 2469, 2613, 2620, 2645, 2679, 2889, 3117, 3123
CTICNT FF2C 845, 846, 1998, 2000, 2027, 2031, 2044, 2070, 2078, 2102,
2104, 2119, 2228, 2489, 2535, 2570, 2593, 2602, 2624, 2961,
3050, 3113, 3124, 3139, 3146, 3608, 3625, 3629, 3638, 3642,
3703, 3707, 3723, 3741
CTICSM FF2A 847, 848, 2561, 2603, 2621, 2632, 3102, 3109, 3111, 3118,
3120, 3134
CTIJMP FFE0 189, 190, 964
CTINTR 283D 932
CTISPT FF31 843, 844, 2461, 2538, 2562, 2587, 2639, 2891, 3088, 3106,
3159
CTISTA FF29 848, 852, 3554, 3585, 3749, 3835
CTITRL FF2B 846, 847, 2466, 2667
CTIVEC FFE1 188, 189, 1460, 1473, 1781, 1886, 2074, 2101, 2512, 2661,
3248, 3685, 3763

```

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'I0273'
SYMBOL VALUE REFERENCED ON

```

=====
CTLALT  4123  5940, 5983
CTLCT   33EF  3514, 3524
CTLLCT  33E8  3510, 4051, 5979
CTLLIM  0020   325
CTLPRT  47AF  7529, 5982
CTLRCT  33FD  3520, 4055, 5980
CTLRED  3784  4155, 899
CTLTAB  414D  5978, 1364, 5663
CTM020  2998  1240, 1242
CTM040  29A9  1250, 1248
CTM100  2A05  1312, 1299
CTMON   296F  1217, 912, 2315, 3933, 4074, 4334, 4443
CTMON1  29C1  1277, 1134, 1221, 1241, 1360, 1819, 1990, 2189, 2306, 2313,
      2381, 2750, 2774, 2787, 2994, 3221, 3563, 3805, 3898, 3977,
      4545, 4993, 5853, 5910, 6540

CTR010  3FAB  5639, 5637
CTR020  3FAF  5646, 5632
CTR025  3FBE  5654, 4359, 5650
CTR030  3FC3  5658, 5673
CTR040  3FD1  5667, 5662
CTRLIO  3F98  5628, 5184
CTSTAT  FF66   713, 724, 933, 937, 939, 945, 961, 1130, 1253, 1313,
      1383, 1903, 3438, 4706, 5726

CTSTRT  2800   891, 2742, 893
CTT100  36FE  4057, 4052
CTUER1  2DED  2329, 7429
CTUER2  2DFF  2341, 2339
CTUERR  2DE5  2325, 1576, 1605, 1638, 2004, 2432, 2814, 2862, 2931, 2972,
      3583, 3889, 4105, 6016

CTUIN   0080   872
CTUTST  36E4  4047, 902, 3991
CURADR  FFC3   555, 557
CURCOL  FFC1   563, 564, 6946, 6975, 7001
CURFKY  FFA4   579, 580
CURPH   0061  7622, 7623, 4257, 4674, 4679, 4696
CURPHD  0064  7623, 7624, 4206
CURROW  FFC0   564, 565, 4677, 6917, 6936, 6992
D       0044   346
D2B010  44CE  6895, 6890
D2B020  44D8  6907, 6893
D2B030  44F3  6923, 6913, 6919
D2B040  44FB  6930, 6916
D2B060  4519  6949, 6926
D2B090  451B  6952, 7006
D2B099  4523  6960, 6969, 6974
D2B100  4525  6963, 6955, 6977
D2B200  4548  6986, 6967
D2B250  456E  7010, 6987
D2B270  457B  7018, 7015
D2B280  4581  7021, 7037
D2B300  458A  7030, 6988, 6997
D2B400  4582  7050
D2B410  4583  7052, 7020, 7044

```


13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'I0273'

SYMBOL VALUE REFERENCED ON

```
=====
D2B420 45C1 7060, 7032, 7054
D2B440 45C5 7064, 6982, 7059
D2B500 45CE 7072, 7026
D2B600 45D8 7085, 7019, 7036
D2BFTB 4143 5969, 6018
DATATR 0040 763, 1180, 1192, 1635, 2179, 2810, 2829, 2904, 2913
DATCOM 0020 791, 4277, 4535, 4917, 5831, 6191, 6255, 6537
DBLHOL 0010 750, 1013, 1053
DC2 0012 322
DC2BUF 40E8 5899, 4283, 5842
DC2SND 0080 70
DC3 0013 323
DCB010 40F9 5908, 5917
DCB020 4102 5914, 5925
DCB030 410B 5918, 5907
DCERR 3CFB 4976
DCERR1 47E2 7594, 5859, 5916
DCHAR FF89 611, 612
DCIOFF 0010 128
DCJMP0 0080 85
DCJMP1 0001 89
DCJMP2 0002 90
DCJMP3 0004 91
DCJMP4 0008 92
DCJMS2 5006 260
DCJMSK 5005 259, 260
DCMCTL 3CF4 4960, 4906
DCMERR 0001 113
DCMNUS 004C 7615, 7616, 4369, 5087
DCNUM 0046 7613, 7614, 4377, 5090
DCPLUS 0049 7614, 7615, 4366, 5085
DCTEST 00A2 7644, 7645, 4019
DECRDX 000A 155, 4324
DEFSKY 0008 107
DELTRM 0000 679
DEVFLG FE7F 870, 879
DFLGS FF6E 666, 676, 3556, 4560, 5350, 6014, 6217, 6565, 6606
DFN020 3EC1 5422, 5415
DFNDEV 3EB0 5410, 5275, 5294, 5309
DFNDV0 3EB3 5412, 4427
DIFMSG 3C3C 4820, 3384
DISCNT 0006 288
DISPLY 0004 788, 4211, 4249, 4490, 4535, 4557, 4702, 5488, 5517, 6631,
6661, 6851, 6880
DISPST FFFE 168, 169
DLRMSG 3C22 4818, 3335
DMAOFF 0060 434
DSP2BF 44B5 6877, 5972
DSPBGN FFAA 576, 577
DSPBTM 0040 740, 6896, 6921, 6980, 7041
DSPEND FFA8 577, 578
DSPFNC 0001 136
DSPLIM FBFF 516
=====
```

13255

2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```
=====
DSPMSG 0040 7611, 7612, 4892
DSPNUM 3C98 4855, 3352, 3360, 3389
DSPSTR FE4F 523, 3386
DSPTYP FFAE 569, 575, 7045
ECONTF FFCD 537, 538, 543, 567
EDA100 3A5E 4663, 4665, 4667
EDA500 3A6E 4673, 4659
EDA520 3A71 4675, 4681
EDA550 3A95 4695, 4688
EDA580 3ABD 4715, 4713
EDA600 3AC2 4718, 4705
EDIT 0010 140, 1236, 2923, 4066, 4204, 4223, 4461, 4467, 4517, 5289
EDR050 3296 3218, 3222
EDRST 3964 4466
EDTWRP 0008 83
ENDATA 3CF2 4952, 6320, 6420, 6507
ENDBAK 2BCC 1834, 3500
ENDBLK 0007 289, 4953
ENDCOL FF21 862, 863, 7002
ENDDSP 0080 741, 6859, 6896
ENDPR 00C1 380
ENDROW FF20 863, 6934, 6995
ENDTST 0006 239
ENHLIM 00BF 378
ENHOUT FF76 635, 636
ENTRCD 0098 7551
EOF 0001 757, 1588, 1933, 1951, 2651, 2913, 2952, 3481, 3533, 3887,
4716
EOFC 2D3D 2153, 3501, 3890
EOFINH 0080 764, 2126, 3481, 3660
EOFMSG 3C15 4816, 3311
EOL 00CC 390, 4576, 4692
EOLADR FF94 598, 599
EOLMV FF90 601, 602
EOP 00CE 391, 4397, 4588, 4792, 4794, 4796, 4801, 4809, 4833, 4836
EOPMSG 38BD 4800, 2343, 3340, 5696
EOTMSG 3B07 4767, 1604, 2930
ERRCHK 47F0 7601, 6190
ERRFLG FFF7 175, 176
ESC 001B 324, 7102
ESCEND 004F 7616, 7617, 5110, 5119, 5124, 5125, 5127, 5130, 5132, 5344
ESCFLG FFD1 201, 204, 5166, 5329
ESCINP 0008 660
EVD 0002 758, 1619, 1635, 1925, 2544, 3016, 3224, 3450, 3533, 3776,
3887
EVDBSP 2D3D 2142, 2886
EVDC 2D69 2196, 2186, 3502
EVDMSG 3B1A 4771, 1621, 1637, 3313, 3778
EVDRED 3278 3204, 4017
EVDWAT 2D5E 2185, 2929
EW 0080 753, 1046, 1602, 1843, 3476, 3893
EXB020 4456 6769, 6798
EXB030 445F 6776, 6773
```

13255

2648A MICROCODE LISTING 'I0273'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
EXB050  4466  6782, 6807, 6823
EXB100  4473  6805, 6778
EXPAND  008E  7637, 7638, 6812
EXPBF0  4437  6746, 6624, 7180
EXPBF1  4439  6748, 6239
EXPBF2  4467  6788, 6270, 6649, 6819
EXPBF3  4468  6795, 6259, 6635, 6820, 7211
EXTB2D  0001   738, 4185, 5780, 6359
F       0046   347, 4628, 6130, 7603
FAILMS  3B68  4786, 3888
FCP100  2D84  2227, 2225
FCTK2D  0010   671
FEVDMS  3B14  4769, 3833
FF      000C   318, 7217, 7541
FILCHR  FF8F   602, 605
FILCM1  2D7D  2223, 3631
FILCMP  2D7A  2221, 3691, 3715
FILL    00C3   382, 4569
FILMSG  3C5B  4824, 3354
FILNUM  FF5E   769, 770, 1701, 1942, 2224, 2649, 2908, 3605, 3740, 5729,
      5736
FILRED  0004   728, 4909, 5347, 5784, 5785, 5793, 6323, 6508
FIVE    0035   339
FLDSEP  00C4   389, 6774, 6932, 6944, 6998
FLINE   FF9F   582, 589
FLINMS  3B42  4780, 5718
FMSMSG  3C6A  4830, 3725, 3743
FMTCTL  FF8A   610, 611
FNCLIM  00A0  7553
FNDTB2  0073  7628, 7629, 5421
FORGN   0080   143
FORMAT  0008   139, 4486, 7575
FOUR    0034   338
FPS     0004   748, 1773, 2860, 3457
FRBLKS  FFAC   575, 576
FRCPTY  0080   99
FRCRST  0004   106
FREBFS  3D17  5031, 1157, 2549, 3965, 4291, 4941, 5377, 5618, 6504
FRECNT  0067  7624, 7625, 4207
FRNMD1  000E   247
FRNMD2  000F   248
FRSALT  4829   229, 230
FRSOUT  0010   661
FRSTBL  FF92   599, 600
FSP020  2C50  1989, 1991, 2016
FST     0004   777, 1705, 1715, 2051, 2113, 3645, 3651, 3729, 3747
FSTBIN  000A   292, 4905
FSTRAM  9100   38, 161, 528
FSTSND  0020   65
FULDUP  0080   42, 4183
FWD     0002   776, 1514, 1557, 1715, 1762, 1938, 2007, 2080, 2491, 2964,
      3012, 3054, 3151, 3216, 3651, 3747
FWDSP.1 2C45  1983, 2685, 2711, 3569

```

13255
 2648A MICROCODE LISTING 'I0273'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
FWDSPX  2C57  1994, 1987, 3541, 3832
GAP      0020  717, 977, 2060, 2090, 2508, 3245, 3255, 3679, 3760
GAPTST  2C15  1922, 2065, 3686
GCT010  2E09  2551, 2530
GCT020  2F00  2572, 2564
GCT030  2F07  2577, 2575
GCT035  2F10  2583, 2571
GCT040  2F25  2595, 2590
GCT050  2F2E  2601, 2610
GCT060  2F3C  2608, 2597
GCT070  2F4D  2617, 2614
GCT100  2F5E  2628, 2559, 2585, 2607
GCT320  2F8F  2654, 2644
GCT420  2FC3  2681, 2671
GEN      0020  780, 1762, 2964, 3012, 3151
GETADR  FF73   637, 641, 7011, 7034
GETCSM  2F62  2631, 2627
GETDAT  2F43  2612, 2606
GETDCM  0094  7639, 7640, 1358, 4664, 6361
GETDSP  0088  7635, 7636, 6965
GETIO   4164  6009, 4914, 6188
GETIO1  4167  6011, 5535, 5611
GETLSB  2F16  2586, 2584
GETMSB  2EE8  2560, 2558
GETPRM  2EE5  2557, 2486, 2533
GETPT1  3D28  5069, 2468, 4565
GETPTR  3D2A  5067, 2888, 3877, 4924, 5710, 5850, 5904, 6193, 6753, 6924,
        6931
GETSTA  2A46  1380, 1278, 1570
GPT010  2C35  1941, 1929
GPT020  2C3F  1947, 1944
GPT030  2C41  1950, 1926
GRN000  3657  3930, 3943, 3953
GRN100  3675  3949, 3952
GRNKEY  365A  3932, 897, 3937
GRNTBL  36A0  3987, 3947
GTCTBT  2ADC  1540, 1128, 1135, 1184, 1302, 1406, 1572, 2156, 2199, 2245,
        2280, 2336, 2539, 2638, 2892, 2985, 3164, 3701, 3873, 4710,
        5724
GTIOB0  3CFF  4992, 2154, 2197, 3865, 4275, 4998, 5708, 5829, 6878
GTIOBF  3D0C  5013, 2458, 4548, 4996
GTMODE  00B1  7649, 4164, 5789, 6310, 6500
H       0048  348
HANGU0  009D  7642, 7643, 4059, 4978, 7595
HCK300  28A1  1025, 1001
HCK400  28B4  1043, 1030
HCK500  28BE  1051
HCK600  28C9  1061, 1054
HNDSHK  0040  67
HOL     0010  718, 941, 1090
HOLCHK  2889  998, 942
HOLCNT  FF51  799, 800, 949, 1000, 1055, 1892
HOLCT0  28D8  1089, 954
  
```

13255
2648A MICROCODE LISTING 'I0273'
SYMBOL VALUE REFERENCED ON

```

=====
HRDER1  0010    761, 2425, 2428, 2683, 2796
HRDERR  0004    759, 2429, 2810, 3450, 3461, 3887
HRDMSG  3B62    4784, 2431
HTBLEN  000A    629, 630
HTBTBL  FF78    630, 634
IGNTRM  0001    680
INDJMP  419A    6090, 3519, 5362, 5375, 5376
INITDG  0085    7634, 7635, 6854
INOMSG  3B77    4790, 6125
INPDEV  FF4E    807, 808, 2302, 3208, 4140, 4488, 4534, 4701, 5636, 5811,
        6010, 6534, 6850

INSCHR  0002    137
INSWRP  0002    105
INTDSO  44A3    6849, 4912, 6181
INTDSP  44A9    6853, 5522
INTFLG  FFF6    176, 177, 7270, 7273, 7305, 7308
INTRWD  2B31    1688, 1730, 1752, 1836
INTVEC  9165    161, 162
INVRS   0082    436, 4760, 4762, 4768, 4770, 4772, 4775, 4778, 4781, 4782,
        4787, 4789, 4796, 4809, 4811, 4813, 4817, 4831, 4833, 4845

IOBASE  0080    417, 421, 429, 441, 450, 457
IOBNUM  0001    5189
IOBUF   FC00    518, 519, 520
IOBUF1  FC00    521, 3367, 5071, 6183, 6198, 6439
IOBUF2  FD00    522, 3368, 5073
IOBUFH  00FC    519, 520
IOBUFL  0000    520
IOC002  3D8D    5160, 5164
IOC010  3D93    5165, 5082
IOC020  3DAD    5210, 5116
IOC030  3DBD    5229, 5117
IOC040  3E01    5274, 5131
IOC050  3DAC    5205, 5120
IOC060  3E0C    5287, 5129
IOC070  3DAB    5200, 5123
IOC080  3E0B    5282, 5118
IOC090  3DF2    5265, 5126
IOC100  3DBB    5223, 5128
IOC110  3E22    5305, 5103, 5105
IOC115  3E32    5314, 5310
IOC120  3DD6    5248, 5133
IOC130  3E4C    5334, 5327
IOC140  3E58    5341, 5337
IOC150  3E85    5367, 5352
IOCCL1  3D8C    5157
IOCCLR  3D87    5145, 3934, 4071
IOCCNT  FFD5    828, 2181, 3539, 3593, 4357, 5253, 5267, 5339, 5653, 5836,
        7532
IOCDEV  FFDB    823, 4347, 5277, 5311, 5629, 5641, 5660, 5808
IOCDPT  FF4C    809, 810, 5379, 5655, 5668, 7147
IOCERR  FF4F    804, 807, 2945, 3573, 3966, 4621, 4627, 5174, 5378, 5952,
        6131, 7602
IOCEX0  3E34    5320, 5218, 5244, 5258, 5260, 5270, 5278, 5300

```

13255

2648A MICROCODE LISTING 'I0273'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

=====

IOCINP	FFD9	825, 4086, 4128, 5292
IOCMND	FFD7	827, 5345, 5356, 5371, 5393
IOCMTB	3D98	5180, 5361, 5370
IOCNTL	3D81	5142, 905
IOCNUM	0004	5192
IOCOU	FFDA	824, 4145
IOCRCL	8700	430
IOCRRW	8720	431, 7543
IOCSGN	FFDD	191, 192, 4349, 4379, 5254, 5268, 5324
IOCT20	3D61	5115, 5093, 5095
IOCT30	3D68	5122, 5098, 5100
IOCTAB	3D32	5079, 5143
IOCTCO	8B00	442, 1008, 1424, 1508, 2268
IOCTDI	8B20	445, 974, 2088, 2270, 2509, 2554
IOCTDO	8B20	444, 3047, 3131, 3145
IOCTSI	8B00	443, 935, 1136, 1224, 1386, 2269, 3153
IOCTU	8B00	441, 442, 443, 444, 445
IOCTYP	FFD8	826, 1356, 3517, 4050, 4189, 4329, 4392, 4441, 5217, 5240, 5450, 5633, 5783, 5787, 6165, 6206, 7535
IODATA	FFDE	190, 191, 696, 3604, 4348, 4388, 5146, 5216, 5237, 5241, 5252, 5266, 5322, 5411
IODISP	8700	429, 430, 431
IODNGO	4615	7144, 907
IOERCL	3D97	5172, 1158, 1256, 1338, 3840, 3844, 3907, 4048, 4160, 4173, 4220, 4562, 4623, 4904
IOFAI0	41AD	6124, 4493, 5505
IOFAI1	4180	6126, 4638, 5485, 5719
IOFAIL	41B3	6129, 2330, 5617, 6572, 7598
IOFLG2	FF64	736, 744, 1156, 4187, 5779, 6358, 6857, 6886, 6978, 7040, 7075
IOFLGS	FF65	724, 736, 1651, 1668, 2546, 2552, 2635, 2698, 6224, 6295, 6316, 6417, 6494
IOFNUM	0002	5190
IOKB	8300	421, 422
IOKBCO	8380	422
IOMNUM	0003	5191
IOPSGN	FFDC	192, 193, 3537, 3594, 4356, 5255, 5269, 5335, 5647
IOPTR1	8D00	450, 451, 452, 453
IOPTR2	8500	457, 458, 459, 460, 461
IOR020	41D3	6176, 6167
IOR025	41EA	6187, 6185
IOR030	41F6	6194, 6179
IOR100	41F9	6196, 6172
IOR110	4204	6202, 6200
IOR210	422C	6235, 6226
IOR220	4238	6245, 6272
IOR230	423E	6250, 6241, 6273
IOR270	4250	6264, 6252
IOR285	4259	6271, 6260
IOR300	4267	6293, 6278
IOR350	4283	6308, 6297
IOR360	428C	6312, 6302
IOR380	428F	6314, 6304, 6448

13255

2648A MICROCODE LISTING 'I0273'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
IOR400  42A7  6343, 6313, 7583, 7592
IOR500  42B2  6349, 6237
IOR550  42B5  6352, 6347
IOR570  42C9  6362, 6357
IORDGO  41BA  6162, 908, 4196, 6319
IOREAD  4067  5777, 5186
IORMGO  00A5  7645, 7646, 5950
IORNUM  0006  5194, 5374
IOS010  40A2  5812, 5810
IOS100  460E  7127, 7113, 7114, 7115
IOS120  4610  7130, 7112
IOSCTL  38B8  4346, 4451
IOSNUM  0007  5195, 5308, 5346
IOSTA0  FF48  813, 814, 5813, 7108
IOSTA1  FF49  812, 813, 5931, 7449, 7471
IOSTA2  FF4A  811, 812, 7457, 7479
IOSTA3  FF4B  810, 811, 3434, 7467, 7491
IOSTAT  4092  5805, 5187
IOSTGO  45E1  7101, 906
IOSTX1  4608  7121, 5990
IOW020  40C8  5849, 5841
IOW023  40CB  5851, 5860
IOW025  40D4  5857, 5864
IOW030  40E3  5868, 5845
IOWNUM  0005  5193, 5251, 5359
IOWRIT  40AB  5828, 5185
IRD010  408A  5792, 5790
JMP     00C3  395
KBDCSW  FFFC  170, 171, 4182
KBDLOK  0040  672
KBFCTK  FF71  642, 644
KBJMP2  FFFA  172, 173, 7172
KBJMP3  FFF9  173, 174
KBJMPR  FFFB  171, 172, 6227, 6298
L       004C  349
LADDR   FFD5  695
LCHAR   FF69  689, 690
LCHKSM  FFD7  697
LCT2BF  2E0B  2364, 5970
LDATA   FFDE  696
LDRCHK  0004  115
LF       000A  317, 4603, 5923, 6350, 6670, 6693, 6829, 6968, 6972, 7065,
        7200, 7381, 7399, 7534, 7591
LFPOS   0010  59, 6228, 6300
LFTBKT  005B  357
LFTBRC  007B  371
LFTCTU  0001  786, 1543, 2303, 2365, 2732, 2940, 2943, 3318, 3321, 3397,
        4478, 4505, 4704, 5508, 5562, 6535
LFTMGN  FFBF  565, 566
LINWRP  0004  55
LITOFF  2A58  1405, 1739
LITUFL  2A5E  1408, 1281
LITOFR  2A63  1411, 1284, 1407

```

13255
 2648A MICROCODE LISTING 'I0273'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
LLINE   FFA1    581,  582
LLPMSG  3ADF    4761, 1815
LNFEED  008B    7636, 7637, 7049
LNKLIM  00D0    392
LNKSAV  FF96    597,  598
LOCKKB  0001    234
LOCMSG  3B37    4777, 3669, 4315
LOF010  2A65    1413, 1410
LOG900  2A2D    1355, 1341, 1343, 1345
LOG910  2A30    1357, 1359, 1361
LOGRWD  2A0C    1335, 1220
LP      0040    752, 1044, 1044, 1064, 1095, 1100, 1769, 1774, 1810, 2033,
        2057, 2086, 3676, 3757
LPM     0001    746, 1063, 1095, 1173, 1257, 1467, 1774, 3476
LPM010  2BC0    1817, 1820
LSTCOL  FFC8    549,  550
LSTDCD  FFC6    553,  554
LSTFMT  FFC5    554,  555
LSTFWD  0002    747, 1774
LSTLIN  FFC9    546,  549
LSTRED  FF25    853,  854, 6168, 6182, 6195, 6437
LSTROW  FFC7    550,  553
LWBUF   00B0    604
LWDSP   00D0    517
M1MSG   3C89    4835, 4327
MANORG  0040    7610, 7611
MAXCOL  004F    401
MAXROW  0017    400,  6918
MAYEOL  0040    433
MAYEOP  0020    432
MDFLG1  FFF4    178,  179, 1235, 2922, 4065, 4203, 4221, 4459, 5288, 7574,
        7596
MDFLG2  FFF3    179,  180, 2769, 3155, 3855, 3862, 4025, 4031, 4156, 4174,
        4239, 4496, 4657, 7588
MEMLOK  0004    138
MESSGE  3AD3    4757
MFLGS   FF70    644,  655
MFLGS2  FF6F    655,  666
MINUS   002D    332, 4368
MLKFLG  FF6A    688,  689, 6590
MLKOFF  009A    7641, 7642, 4494, 4669, 4686
MLKROW  FF6B    687,  688
MLKSCH  0097    7640, 7641, 4687
MNSMSG  3C90    4841, 4385
MSGPT1  FFF1    180,  181, 2335, 3315, 3336, 3385, 4058, 4326, 6127
MSGPT2  FFEF    181,  182, 2342, 3325, 3338, 3387, 4328, 4387, 5711
MSGPT3  FFED    182,  183, 2344, 3341, 3362, 4391, 5691
MSGPT4  FFEB    183,  184, 3358, 5693
MSGPT5  FFE9    184,  185, 3342, 3355, 5716
MSGPT6  FFE7    185,  186, 3350, 5695
MSGPT7  FFE5    186,  187, 3343, 5697
MSGPT8  FFE3    187,  188
N       004E    350
  
```


13255
2648A MICROCODE LISTING 'I0273'
SYMBOL VALUE REFERENCED ON

```

=====
NBLKS      FF99      594, 595
NCHAR      FF9B      592, 593
NEWCOL     FFDB      622
NEWROW     FFDA      623
NLTPMS     3BBE     4802, 5690
NMFCTK     0008      425
NMROLL     FF83      619, 620
NODCST     U010      93
NOFNCT     4163     5995, 5981
NORMAL     0080      437
NOSEND     0004      670
NOSIGN     0080      409
NOTEST     0004      82
NRCMSG     3B27     4774, 2861
NROWS      FF9A      593, 594
NRTPMS     3BCE     4805, 5694
NTPMSG     3B6E     4788, 1575
NULL       0000      316
NULMSG     3C14     4814, 3337
NUM2K      0800      393
NUMBER     00C6      385
NUMSWP     000F     567, 568
NWRWST     0080      664
NXTCHR     0091     7638, 7639, 4567, 4580
NXTRED     FF27      852, 853, 6177, 6294, 6390, 6506
OCM000     2A8B     1472, 1706, 1716, 1763, 1879, 2081
OCM001     2A8E     1474, 1902, 2114, 5749
OCM005     2A92     1478, 1471
OCM010     2AAA     1492, 1490
OCM020     2A84     1499, 1497
OCM030     2ABE     1506, 1487
OCTRDX     0008      156
OFFMSG     3AF4     4763, 1118
QLTPMS     3BBF     4803, 2338
OPSTOR     FFD0      529, 530, 537
ORTPMS     3BCF     4806, 2340
OTHER      FF56      771, 773, 2251, 2252, 4714
OUTCM1     2A7B     1461
OUTCMD     2A78     1459, 2050, 2052, 2492, 2966, 3014, 3055, 3152, 3217, 3665,
          3731, 3803
OUTDEV     FF4D      808, 809, 2938, 4148, 4210, 4248, 4476, 4502, 5470, 6042
PAGSTR     0008      57
PARM1      FFDB      193, 194, 622, 823
PARM2      FFDA      194, 195, 623, 824
PARM3      FFD9      195, 196, 624, 825
PARM4      FFD8      196, 197, 826
PARM5      FFD7      197, 198, 697, 827
PARM6      FFD5      198, 199, 695, 828
PCT020     31DA     3096, 3094
PCT030     31E1     3101, 3099
PCT100     3219     3128, 3078, 3082, 3086, 3104, 3115, 3136, 3142
PCT200     321C     3130, 3126
PERIOD     002E      333

```

13255

2648A MICROCODE LISTING 'I0273'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```
=====
```

PI0010	4181	6047, 6060
PLSMMSG	3C8D	4838, 4383
PLUS	002B	330, 4365
POLL	0040	130
PRCCTL	FFF5	177, 178
PRCHR1	4686	7253, 7347
PRCHR2	4685	7284, 7364, 7395
PREMSG	3B8B	4795, 7428
PRINTR	0008	789, 4137, 7225
PRNTAL	0010	84, 7173
PROCSR	0070	416
PROFLD	FFC2	557, 563
PROMPT	000D	295
PRTX05	47C0	7539
PTBLK0	006A	7625, 7626, 4666, 4680
PTDLY	05DC	498, 7254, 7285
PTP002	39B3	4533, 4555
PTP005	39DA	4556, 4549, 4552
PTP010	39EB	4566, 4570, 4572, 4577
PTP020	3A06	4579, 4581
PTP030	3A0C	4585, 4568
PTP040	3A18	4593, 4589
PTP050	3A21	4600, 4597
PTP220	3A42	4625, 4592, 4620, 4639
PTP490	3A4C	4634, 4575
PTP500	3A4D	4636, 4602
PTR110	4689	7255, 7272, 7278
PTR120	46A1	7269, 7261
PTR630	46B8	7286, 7307, 7313
PTR635	46D0	7299, 7295
PTR640	46D5	7304, 7291, 7298
PTR700	4749	7427, 5973, 7258, 7277, 7312
PTRABT	FE78	882, 885, 7417, 7430, 7454
PTRBBG	FE7D	879, 880
PTRBD2	001F	511, 7489
PTRBLN	0100	524
PTRBPT	FE79	881, 882
PTRC30	46FB	7343, 7339, 7350
PTRC50	4704	7360, 7335
PTRC60	4706	7363, 7403
PTRC70	4728	7383, 7379
PTRC80	4730	7392, 7382, 7388
PTRC85	4732	7394, 7398
PTRC90	473C	7400, 7372, 7377
PTRCF2	8540	461, 7293, 7368, 7487
PTRCHK	4741	7416, 7165, 7530
PTRCHR	46E9	7332, 7202, 7242, 7538, 7544
PTRCL1	8D02	453, 7267
PTRDA2	8560	460, 7302
PTRDY1	0001	502
PTRDY2	0002	507, 7290, 7495, 7496
PTRFLG	FE77	885, 7333, 7458
PTRHD2	00E0	510, 7294, 7369, 7371, 7502

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'I0273'

SYMBOL VALUE REFERENCED ON

=====

PTRINP	0020	877	
PTR0L2	0020	509, 7495	
PTR0T1	8D20	451, 7263	
PTR0T2	8540	458	
PTRP01	0080	503	
PTRS10	4784	7477, 7466, 7497, 7506	
PTRS20	4781	7473, 7470	
PTRS80	478B	7486, 7460	
PTRS82	0040	508, 7297, 7505	
PTRSPT	FE7B	880, 881	
PTRST1	8D00	452, 7256, 7336, 7464	
PTRST2	8520	459, 7287, 7296, 7493	
PTTPLN	39C0	4543, 913, 4694	
PTTPOK	0040	792, 4550, 6256, 6632, 6661	
PUTBRK	0005	287	
PUTCSM	3220	3133, 3127	
PUTDAT	3200	3116, 3114	
PUTIO	417A	6041, 4308, 4619, 4624, 5871	
PUTI01	417D	6043, 5597	
PUTLSB	31EA	3105, 3103	
PUTMSB	31CD	3087, 3085	
PUTP01	3229	3137, 3135	
PUTPOS	3236	3143, 3140	
PUTPR2	31B7	3075, 2962	
PUTPR3	31BE	3079, 3077	
PUTPR4	31C5	3083, 3081	
QUOTE	0027	328	
R	0052	351	
RADIX	FFD4	199, 200, 4325	
RCADRA	00AE	7648, 7649	
RCK650	3845	4274, 4289	
RCK700	3860	4290, 4236, 4276, 4300, 4302	
RCK750	386B	4297, 4286	
RCK800	3876	4304, 4287, 4301	
RCRDGO	383B	4269, 909	
RCT020	2937	1171	
RCT100	2964	1191, 1179	
RCT120	2969	1194, 1167, 1174	
RCT2BF	2E10	2367, 5971	
RCVMDE	0020	109	
RDA005	4383	6539, 6541	
RDA010	4398	6549, 6547	
RDA020	43A0	6554, 6552	
RDA030	43A3	6559, 6536	
RDABR1	4375	6532, 4911, 5349	
RDABRT	4375	6531, 915, 6249, 6416, 7605	
RDI020	2E63	2433, 2426	
RDINIT	2E3F	2414, 2380, 2392	
RDNEXT	2E67	2454, 2663	
RDRTRY	2FA7	2666, 2534, 2568, 2582, 2592, 2599, 2611, 2616, 2634	
RDSTRT	2E85	2485, 2680	
RDVERF	2E7B	2465, 2785	
RDOWWT	0001	726, 4165, 4195, 4909, 5347, 5791, 6318, 6418, 6508	

13255

13255/90010

2648A MICROCODE LISTING 'I0273'

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
RDY      0040      716, 3071
REC      0008      778, 1166, 1298, 1762, 2964, 3012, 3054, 3151
RECINI   0010      731, 4237, 5520, 5900
RECKEY   37EF     4219, 900, 1239
RECMSG   3C51     4822, 3361
RECORD   0040      142, 4235, 4241, 4271, 4292, 7597
RECPGE   0020      732, 5527, 5559, 7042
RECRWD   0008      729, 1218, 2936, 4474
RECSEP   5003      257, 258
RECSPC   2CA2     2043, 2008
RED010   37AC     4181, 4167
RED020   37B6     4186, 4184
RED100   37D4     4202, 4177
RED120   37E3     4209, 4205
REDKEY   379F     4172, 898, 4159
RELSNS   0004      659
RELTAK   FF61     766, 767, 979, 1776, 1791, 1873, 1927, 1985, 2036, 2069,
                2091, 2655, 2958, 3006, 3149, 3644, 3688, 3728, 3770
REMOTE   0008      150, 4158, 4175, 4176, 4240, 4497, 4658
REMSET   0010      108
REMSPC   340C     3532, 3504
RESET    0000      414
RET      00C9      396
RETSCO   4831      28, 29, 4994, 5911
RETSCN   482E      27, 28, 1818, 2311, 2455, 3219, 3566, 3978, 4678, 5854
REV010   2D62     2188, 2190
REVEVD   2D52     2177, 1694, 3210, 3531, 3640
REXMIT   0001      406, 6166
RGTCTU   0002      787, 1545, 2303, 2368, 2735, 2940, 2943, 3321, 3401, 4478,
                4505, 4509, 4704, 5508, 5562, 6535
RHTMGN   FFBE     566, 567, 568
RIP      0004      720, 1773
RLP100   2B7E     1760, 1814
RNGTA    FFD2      200, 201, 5144
ROLLCT   FF82      620, 630
RPTKEY   0003      236, 4106
RSETDC   0002      284
RSETKB   0007      240
RSP040   2CC4     2067, 2125
RSP110   2CF7     2095, 2092
RSP120   2D1B     2118, 2109
RSTCTU   291B     1154, 904
RSTDSP   0043     7612, 7613, 1139, 1821, 2307, 2312, 2712, 3564, 3572, 3980,
                4340, 4358
RSTJMP   0001      415, 3960, 5546
RSTOFF   0004      424
RSTON    0002      423
RSTTMR   FFD0      204
RTRYMS   3BB7     4798, 2705
RUN      0001      775, 1287, 1382, 1485, 1496, 1705, 1715, 1762, 1878, 1899,
                2007, 2037, 2080, 2416, 2491, 2853, 2964, 3012, 3054, 3151,
                3216, 3645, 3651, 3729, 3747, 3801
RWDBOT   2B66     1728, 1848, 3496

```

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'I0273'

SYMBOL VALUE REFERENCED ON

```

=====
RWDLP 2875 1750, 1195, 1259, 3627, 3896
S 0053 352, 3574, 3967, 5173
SAVINP FF23 857, 858, 5516, 5534, 5561
SAVOUT FF22 858, 862, 5502, 5596, 5610
SBINRY 0002 658, 5368, 6421, 6454, 6561
SBLXF0 0058 7619, 7620, 5786, 5795, 5807, 6324, 6422
SBLXFA 005B 7620, 7621, 4198, 5381
SCNCNT FF54 795, 796
SCNVEC 9168 162
SCRNRW FFD9 624
SCRSEN 1000 650
SDACOM 0001 668, 3557, 5351
SDAULF 47D7 7587, 6230
SDBYCT 42CC 6388, 6208
SDC2 0100 646, 5368, 7122
SDCRLF 47D2 7581, 6303, 6309, 6502
SDEOF 4345 6493, 6171, 6205, 7604
SDEOF1 435A 6503, 6560
SDTERM 0076 7629, 7630, 7117, 7149
SDTRM1 0079 7630, 7631, 6415
SDVDUN 8000 653, 5368, 5380, 7145
SDVREC 0001 657, 4197, 5368, 5778, 6163, 6321, 6454, 6561
SDVST 0800 649, 5806, 7122
SEARCH 3490 3622
SEF100 4357 6501, 6497
SELACT 2D89 2238, 1255, 2419, 2856, 3209, 6543
SELECT 0020 141, 4068, 4109
SELKEY 3704 4064, 901
SELLCT 2D8D 2244, 3513, 5698
SELOPP 2D91 2247, 2282
SELRCT 2DBA 2279, 2240, 3523, 5714
SENER 4000 652
SETCH 0020 95
SETCT0 3272 3184, 1193, 1301, 2127, 2652
SETCTW 3270 3173, 2752
SETDEV 3764 4127, 4104, 5342
SETFRN 000C 245
SETJMP 4194 6076, 1365, 5664, 5815, 6019, 6052
SETLCL 0004 286
SETMON 0008 290
SETNRM 0009 291
SETREM 0003 285
SETROM 0080 132
SETTRG 0001 283
SEVEN 0037 341
SFCTKY 2000 651
SFKYAT 00C8 387
SFTCNT FF5D 770, 771, 1693, 2250, 2252, 2701, 3410, 4714
SFTCR 00EF 30, 3939, 4338, 4447
SFTDLY 0032 408
SFTEND 0010 402
SFTERR 0008 760, 2428, 2672, 2682, 2809, 2825, 3450, 3461, 3887
SFTKYS FFA6 578, 579

```

13255

2648A MICROCODE LISTING 'I0273'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```
=====
```

SI	000F	321
SIX	0036	340
SKPMSG	3AD3	4759, 3547, 4322
SLANT	002F	334
SLK050	3718	4073, 4077, 4081, 4099
SLK130	374A	4093, 4090
SLK150	374B	4095, 4088, 4092
SLK200	3751	4103, 4083
SLK410	375C	4108, 4079
SLKYCD	009E	7552, 4078
SLTPM1	2DF6	2337, 3319
SLTPMS	2DF0	2334, 3548, 3670, 3834, 4871
SMALLA	0061	363
SMALLD	0064	365
SMALLF	0066	366
SMALLI	0069	367
SMALLK	006B	368
SMALLP	0070	369, 7106
SMALLX	0078	370
SNDATN	000B	293
SNDFCT	000C	294
SO	000E	320
SPC002	3425	3543, 3540
SPC005	3428	3545, 3542
SPC010	343F	3562, 3567
SPC020	3454	3571, 3565
SPCEN1	345F	3582, 3726, 3744, 3779
SPCEND	3462	3584, 3746
SPCPFL	346A	3592, 3498
SPCPRC	3407	3529, 3497
SPCWAT	342F	3553, 3654
SPCWT1	343A	3559, 3836
SPF100	347F	3602, 3596, 3598, 3600
SPLDIS	0002	53
SPOWL	FF6C	682, 687
SPOWOF	00FF	685
SPOWON	0020	684
SRC050	34AF	3636, 3633
SRC070	34C5	3647, 3632
SRC080	34CE	3652, 3646
SRC100	34D4	3659, 3653
SRC110	34DA	3663, 3748
SRC310	3513	3700, 3714, 3716
SRC320	3533	3718, 3708
SRC500	354D	3735, 3677, 3758
SRC600	3585	3769, 2061, 3680
SRC700	359A	3785, 3694, 3777
SRC720	358C	3804, 3806
SREVD	35D0	3830, 3499
SRT500	2FE1	2707, 2700, 2703
SSTAT	0200	647
SSTAT2	0400	648
STACK	9160	528

```
=====
```

13255
 2648A MICROCODE LISTING 'I0273'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
STALMS 3C63 4828, 1295
STALT 4115 5930, 5992
STATTB 4157 5987, 5814
STBLMD 0004 237
STC010 338A 3413, 3409
STC020 33A0 3447, 3440
STC030 33BB 3468, 3462, 3466
STC040 33CC 3484, 3482
STCHST 000D 246, 6953
STCRLF 4493 6824, 6763
STCT 337A 3403, 3399
STD010 377A 4144, 4131
STIOFS 2824 1650, 2781, 2937, 4166, 4194, 4238, 5528, 5794
STLCT 3370 3396, 5988
STNDSP 44AD 6856, 7091
STOPTH 2BE4 1874, 1124, 1161, 1293, 1713, 1738, 1794, 1845, 2076, 2122,
2543, 2659, 2665, 2674, 2921, 2984, 3034, 3157, 3169, 3220,
3568, 3695, 5563

STP100 2BF0 1881, 1884
STP200 2C0C 1900, 1898
STPFLG 00C4 383, 4571
STPR 00C0 379
STPRPT 0009 242
STPRT 4755 7447, 5991
STPTP0 2BE0 1871, 2058, 2087, 2637, 3257, 3736, 3825
STPTP1 2BE7 1876, 3790
STPXFR FFFF 678
STRCT 3377 3400, 5989
STRTAK 41B0 768, 1778, 5752
STRTBL 005E 7621, 7622, 4247
STRTRY 2FCB 2697, 2675, 2820
STRTST 0005 238
STUNTO 289B 1017, 1047, 1096, 1703, 3516, 3842
STWBSR 36D4 4024, 3505, 3858, 4015
SWCHAR 000B 244
SWPCTU FF24 854, 857, 1237, 1347, 1363, 2924, 2941, 4469, 4515
SWPSTR FFAF 568, 569
T 0054 353
TAK 0008 719
TCHAR FF68 690, 691
TCT005 3608 3864, 3859, 3901
TCT030 361F 3879, 3883
TCT050 363F 3895
TCT060 3642 3897, 3900
TEMP FF9D 590, 591
TEMP1 FF9E 589, 590
TEST 007F 7632, 7633, 4053, 4054, 4056
TESTKY 00FA 7554
TESTOK 0002 114
THREE 0033 337
TID00 287D 973, 914, 1885, 1904, 2660, 3150
TIEBK 2BD2 1842, 1835
TIEDRO 32AB 3243, 3215
  
```

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'I0273'
SYMBOL VALUE REFERENCED ON

```

=====
TIEDR1  3286  3253, 3247
TIGCT0  2E9A  2506, 2490
TIGCT1  2EA8  2526, 2511
TIPCT0  319C  3045, 2965
TIPCT1  31AF  3069, 3053
TIRLP0  2B7B  1757, 1751
TIRLP1  2B86  1768, 1761
TIRLP2  2BA3  1787, 1780
TIRSP0  2CB6  2056, 2049, 2079, 2100
TIRSP1  2CE4  2085, 2073
TIRWB1  2B6C  1735, 1729
TIRWD   2B58  1710, 1704
TISRC0  34E6  3675, 3664, 3730, 3762
TISRC1  3575  3756, 3684
TISRC2  35C3  3814, 3788, 3802
TISTOP  2BFE  1891, 1875
TIWEVD  318F  3024, 3013
TKI     0080  715, 977
TLINO   FFA3  580, 581, 6938, 6993
TMFMSG  3BDF  4808, 4134
TMIACK  0000  124
TMIEN   0002  127
TMIOFF  0020  129
TMPCOL  FF85  614, 618
TMRINT  0003  119, 7271, 7306
TMRON   0001  126
TMTMSG  3B94  4793, 5480
TOPLIN  FFCB  543, 546
TPS100  4020  5723, 5712, 5717
TPS120  4034  5735, 5728
TPS130  4048  5750
TPS140  4054  5757, 5755
TPS160  405A  5760, 5763
TPSTAL  FF50  800, 804, 943, 1288, 1509, 1882
TPSTAT  3FDB  5687, 4021
TRIGGR  5002  256, 257
TRMFCT  FF6D  676, 682
TRMTYP  FFFD  169, 170
TSTCTU  35F7  3854, 3503
TWO     0032  336
U       0055  354, 3969, 6571
UETMSG  3AFC  4765, 1121
UNIT0   FF63  744, 755, 963, 1018, 1027, 1172, 1465, 1512, 1601, 1775,
      1809, 2032, 2326, 2328, 2859, 3408, 3892
UNLKKB  0002  235
USL     0010  779, 1463, 1476, 1481, 1488, 1542, 2266, 3398, 3406
USRCMA  32BF  3277, 3989
USRCMF  32BE  3275, 4011
USRCMR  32BD  3273, 4009
USREAD  0002  727, 4193, 4909, 5347, 6225, 6296, 6323, 6496, 6510
USREDA  3A56  4656, 1339, 4233
USREOF  393E  4438, 3997
USREXT  3684  3964, 1178, 1188, 1196, 1222, 1260, 1350, 3588, 3910, 4111,

```


13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING '10273'

SYMBOL VALUE REFERENCED ON

```

=====
                                4214, 4254, 4256, 4259, 4294, 4546, 4630, 4722, 5363, 6513
USRFFL  387C  4314, 3993
USRINT  4834      29, 6533, 7599
USRNPO  3940  4440, 3909, 4434
USRNPM  3943  4442, 4445, 4453
USRRWD  393A  4432, 3999
USRSKP  3883  4321, 3995
USRTED  395B  4458, 4001, 4703, 4708, 4717, 4721
USRXFA  3ACF  4740, 4007
USRXFF  3ACE  4734, 4005
USRXFL  3ACD  4728, 4003
USRXFR  3ACF  4749
USS010  3886  4323, 4317
USS020  3898  4331, 4342
USS030  389B  4333, 4336, 4345
USS050  38C7  4355, 4351
USS300  38D3  4364, 4341
USS305  38F0  4378, 4367, 4370
USS310  3903  4386, 4382, 4384
UTE030  397D  4484, 4462
UTE050  39AA  4512, 4507
UTE070  39AB  4514, 4498
UTE100  39AE  4516
UTX100  3694  3976, 3979
VERIFY  0080   734, 2547, 2553, 2636, 2699, 2780, 2790
VERSN   0054    3, 892, 2744, 4229, 5704
VRTBAR  007C   372
WBSR    0020   151, 2770, 3156, 4027, 4033
WREOF   30C9  2907, 2901
WREVD   3138  2970, 2902
WRFMSG  3C08  4812, 2813
WRINIT  3085  2851, 2749, 2821
WRNORM  30C2  2903
WRPDEL  0020   662
WRPFLG  0040   663
WRS020  30DA  2917, 2906
WRS025  30FD  2935, 2926
WRS030  3114  2949, 2916, 2919
WRS040  311F  2956, 2954
WRS050  3160  2993, 2996
WRS060  316D  3000, 2975
WRS070  318D  3018, 2973, 2992, 2995
WRSTRT  30A2  2885, 3170
WRTERR  0020   762, 1300, 2757, 2810, 2825, 2829, 3174, 3465, 3481, 3483,
                                3887
XBF2DS  0080   673, 4561, 6015, 6219, 6564, 6608
XCH050  2D9B  2253, 2262
XCHINT  43BB  6588, 6360, 6676
XDS2BF  0020   739, 6888, 6896, 7077
XFR001  3ECE  5472, 4213, 4252
XFR050  3EEC  5493, 5489
XFR100  3EF4  5500, 5475
XFR200  3F26  5532, 5518, 5554

```

13255

2648A MICROCODE LISTING 'I0273'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
XFR220  3F39  5543, 5590
XFR250  3F42  5551, 5583
XFR500  3F4C  5558, 5537, 5547
XFR600  3F5C  5574, 5542
XFR650  3F6C  5584, 5576
XFRD2D  3ECA  5469, 3279, 4720, 4751
XFREOF  3EC5  5442, 5182
XFREVD  3EC6  5449, 5183
XFRFCN  3F76  5595, 5473
XFR LIM  FF47   814, 815, 1183, 2388, 2657, 3697, 4190, 5501, 5781
XFRREC  3EC4  5435, 5181
XMONLY  00C2   381
XPUTD3  007C  7631, 7632, 4938, 4940, 6348, 6356, 6392, 6402, 6410, 6414,
        6463, 6499, 7103, 7105, 7107, 7132, 7148
XTRASP  FE80   868, 870
Y        0059   355
Z        005A   356
ZALPCK  4823   224, 225
ZBELL   4814   219, 220, 1125, 3931, 4067, 4344, 4452, 4963
ZBRK1   3000  2743, 4227, 2745
ZBRK1C  3002  2746, 2741
ZBRK2   3800  4228, 5702, 4230
ZBRK2C  3802  4231, 4226
ZBRK3   4000  5703, 5705
ZBRK3C  4002  5706, 5701
ZCHKTK  6070   32, 6619, 6688
ZCLMD1  4811   218, 219, 4110, 4293, 4468
ZCLXMT  481A   221, 222
ZCTLAL  9214   309, 310, 5941
ZDCBAS  5000   255, 256, 266
ZDCCTL  5011   269, 270, 4961
ZDCINT  5026   276
ZDCMON  500E   268, 269
ZDCTST  5014   270, 271
ZERU    0030   335, 4371, 4373
ZGETAL  920E   307, 308, 5947
ZGETDC  5017   271, 272, 5915
ZGETKY  4805   214, 215, 3936, 4075, 4335, 4444
ZGGTST  6061   31, 6912, 7031
ZGTBIN  501D   273, 274, 5858
ZIN2AL  9205   304, 305
ZIN2DC  500B   267, 268
ZINIAL  9202   303, 304
ZINIDC  5008   266, 267
ZINIKB  4802   213, 214
ZINTAL  9208   305, 306
ZKBBAS  4800   212, 213
ZKBCTL  4808   215, 216, 4107, 6954
ZKBMON  480B   216, 217
ZMONAL  920B   306, 307
ZMSGAL  921A   311, 3324
ZNDBIN  5023   275, 276, 4942, 6447, 6530
ZNUMCK  4826   225, 229

```

13255

2648A MICROCODE LISTING 'I0273'

SYMBOL VALUE REFERENCED ON

13255/90010

REV 04/17/78

=====

ZPUTAL	9211	308,	309,	5944
ZPUTDC	501A	272,	273	
ZSTAAL	9217	310,	311,	5938
ZSTBIN	5020	274,	275,	6444
ZSTJPR	481D	222,	223	
ZSTLKY	4820	223,	224	
ZSTMD1	480E	217,	218,	4070, 4243, 4273, 4518
ZSTXMT	4817	220,	221	
ZZZZZZ	4FFD	7560		

1178 SYMBOLS, 3725 REFERENCES, -50 WORK TRACKS



ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
1	0000	.	ASB,HEX ;KG14 26MAY77	1
2	0000	.	;*****	
3	0000	.	; MODIFICATIONS TO KY36C FOR GRAPHICS	
4	0000	.	; 1. ENTRY VECTORS TO GRAPHICS ADDED.	
5	0000	.	; 2. ADD DFAD LWRASC TO KEYBOARD ENTRY VECTORS.	
6	0000	.	; 3. MODIFY UPPER AND LOWER ASCII TABLES TO USE	
7	0000	.	; NUMERIC PAD FOR GRAPHICS FUNCTIONS.	
8	0000	.	; 4. IN GTK130, CHECK FOR GRAPHICS CURSOR KEY	
9	0000	.	; RELEASED.	
10	0000	.	; 5. IN GTK200, CHECK FOR GRAPHICS CURSOR KEY	
11	0000	.	; PRESSED.	
12	0000	.	; 6. IN KBMON, CALL GRAPHICS TIMER INTERRUPT	
13	0000	.	; ROUTINE.	
14	0000	.	; 7. ASCII TABLES MODIFIED FOR GRAPHICS KEYPAD	
15	0000	.	; FUNCTIONS. GRAPHICS CODES ARE IN RANGE	
16	0000	.	; 207B TO 227B, AND 241B TO 245B FOR CURSOR.	
17	0000	.	; 8. TEST FOR GRAPHICS FUNCTION IS MADE IN SETRPT.	
18	0000	.	; 9. CLEAR TEK MODE ECHO SUPPRESS IN BELL	
19	0000	.	; 10. CODE FOR (SOFT) RETURN KEY NOW 357B	
20	0000	.	; 11. TEST FOR TEK MODE STRAPS MADE IN SET	
21	0000	.	; JUMPER ROUTINE	
22	0000	.	; 12. REPEAT DELAY FOR ZOOM KEYS SLOWER	
23	0000	.	; 13. ROUTINE FROM I/O CODE MOVED TO KEYBOARD	
24	0000	.	; SPACE	
25	0000	.	;*****	
26	0000	.	;	
27	0000	.	; COMMON EQUATES - CM34 - 6/10/76 - 1315 HRS.	
28	0000	.	;	
29	9100	.	FSTRAM EQU 110400Q ;FAST RAM LOWER LIMIT	
30	0000	.	;*****	
31	0000	.	; KBDCSW - KEYBOARD DATA COMM SWITCHES *	
32	0000	.	;*****	
33	0080	.	FULDUP EQU 200Q ;HALF/FULL DUPLEX	
34	0000	.	;*****	
35	0000	.	; KBJMPR - KEYBOARD INTERFACE JUMPERS *	
36	0000	.	;*****	
37	0000	.	;	
38	0000	.	; JUMPERS SENSED AS 0' WHEN INSERTED	
39	0000	.	;	
40	0000	.	; ALL JUMPERS ARE NORMALLY INSERTED	
41	0000	.	;	
42	0001	.	CONDIS EQU 001Q ;CONTROL CODE DISABLE	
43	0000	.	; (0=DISABLED)	
44	0002	.	SPLDIS EQU 002Q ;SPOW LATCH DISABLE	
45	0000	.	; (0=DISABLED)	
46	0004	.	LINWRP EQU 004Q ;COLUMN 80 AUTO CR,LF	
47	0000	.	; (0=ENABLED)	
48	0008	.	PAGSTR EQU 010Q ;PAGE MODE STRAP	
49	0000	.	; (0=LINE-FIELD MODE)	
50	0010	.	LFPOS EQU 20Q ;LINE FEED POSITION	

```

=====
ITEM      LOC   OBJECT CODE  SOURCE STATEMENTS                                PAGE  2
=====
51      0000   . . .      ;                                (0 = POSITION LINE FEED
52      0000   . . .      ;                                AT START OF NEXT I/O
53      0000   . . .      ;                                READ
54      0000   . . .      ;                                1 = PUT LINE FEED AT END
55      0000   . . .      ;                                OF RECORD)
56      0020   . . .      FSTSND EQU 40Q      ;9600 BAUD DATACOM SHIFT
57      0000   . . .      ;                                (0=9600 BAUD FOR ESC,E)
58      0040   . . .      HNDSHK EQU 100Q    ;BLOCK TRANSFER HANDSHAKE
59      0000   . . .      ;                                (0 = FOLLOW DC2SND SETTING
60      0000   . . .      ;                                1 = SEND DC2 BEFORE DATA)
61      0080   . . .      DC2SND EQU 200Q
62      0000   . . .      ;                                (0 = SEND DC2 ON ENTER
63      0000   . . .      ;                                AND FUNCTION KEY IN
64      0000   . . .      ;                                BLOCK MODE
65      0000   . . .      ;                                1 = INHIBIT ALL DC2
66      0000   . . .      ;                                HANDSHAKE)
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	3
68	0000	. . .	;*****		
69	0000	. . .	; KBJMP2 - SECOND SET OF KEYBOARD JUMPERS *		
70	0000	. . .	;*****		
71	0001	. . .	AUTTRM EQU 1Q ;AUTO TERMINATE ON "ENTER"		
72	0002	. . .	CLRTRM EQU 2Q ;CLEAR TERMINATOR ON TRANSMI		
73	0004	. . .	NOTEST EQU 4Q ;INHIBIT TERMINAL SELF-TEST		
74	0008	. . .	EDTWRP EQU 10Q ;INVERT SENSE OF EDIT WRAP		
75	0010	. . .	PRNTAL EQU 20Q ;SEND ALL CODES TO PRINTER		
76	0080	. . .	DCJMP0 EQU 200Q ;DATA COMM JUMPER		
77	0000	. . .	;*****		
78	0000	. . .	; KBJMP3 - THIRD SET OF KEYBOARD JUMPERS *		
79	0000	. . .	;*****		
80	0001	. . .	DCJMP1 EQU 1Q ;DATA COMM JUMPERS		
81	0002	. . .	DCJMP2 EQU 2Q ;.		
82	0004	. . .	DCJMP3 EQU 4Q ;.		
83	0008	. . .	DCJMP4 EQU 10Q ;.		
84	0010	. . .	NODCST EQU 20Q ;INHIBIT DATA COMM SELF-TEST		
85	0000	. . .	; (0 = DISABLED)		
86	0020	. . .	SETCH EQU 40Q ;TURN ON "CH" CONTROL LINE		
87	0000	. . .	; (0 = OFF, 1 = ON)		
88	0040	. . .	CHEKCC EQU 100Q ;MONITOR CC CONTROL LINE		
89	0000	. . .	; (1 = ENABLED)		
90	0080	. . .	FRCPTY EQU 200Q ;FORCE PARITY ON/NO IN CHECK		
91	0000	. . .	; (1 = ENABLED)		
92	0000	. . .	;*****		
93	0000	. . .	; CMFLGS - COMMON FLAGS *		
94	0000	. . .	;*****		
95	0001	. . .	BLKTRG EQU 1Q ;BLOCK TRANSFER TRIGGER		
96	0002	. . .	INSWRP EQU 2Q ;INSERT WITH WRAP AROUND		
97	0004	. . .	FRCRST EQU 4Q ;FORCE FULL TERMINAL RESET		
98	0008	. . .	DEFSKY EQU 10Q ;DEFINE SOFT KEY MODE ENABLE		
99	0010	. . .	REMSET EQU 20Q ;REMOTE MODE ENABLED		
100	0020	. . .	RCVMDE EQU 40Q ;TERMINAL IN RECEIVE MODE		
101	0000	. . .	;*****		
102	0000	. . .	; ERRFLG - ERROR FLAGS *		
103	0000	. . .	;*****		
104	0001	. . .	DCMERR EQU 1Q ;DATACOM (1 = ERROR)		
105	0002	. . .	TESTOK EQU 2Q ;SELF-TEST (0 = ERROR)		
106	0004	. . .	LDRCHK EQU 4Q ;LOADER CHECKSUM (0 = ERROR)		
107	0000	. . .	;*****		
108	0000	. . .	; INTFLG - INTERRUPT FLAG *		
109	0000	. . .	;*****		
110	0003	. . .	TMRINT EQU 3 ;TIMER INTERRUPT		

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE   4
=====
112      0000      . . .      ;*****
113      0000      . . .      ; PRCCTL - PROCESSOR CONTROL FLAGS *
114      0000      . . .      ;*****
115      0000      . . .      TMIACK EQU  0Q      ;ACKNOWLEDGE TIMER INTERRUPT
116      0000      . . .      ;
117      0001      . . .      TMRON  EQU  1Q      ;SET TIMER ON
118      0002      . . .      TMIEN  EQU  2Q      ;RE-ENABLE TIMER INTERRUPT
119      0010      . . .      DCIOFF EQU  20Q     ;DISABLE DATA COMM INTERRUPT
120      0020      . . .      TMIOFF EQU  40Q     ;DISABLE TIMER INTERRUPTS
121      0040      . . .      POLL   EQU  100Q    ;POLL CTU INTERRUPTS
122      0000      . . .      ;V*V*V*V* SET TO ZERO FOR ROM VERSION *V*V*V*V*
123      0000      . . .      SETROM EQU  0      ;0 = ENABLE ROM
124      0000      . . .      ;*****
125      0000      . . .      ; MDFLG1 - TERMINAL MODE FLAGS 1 *
126      0000      . . .      ;*****
127      0001      . . .      DSPFNC EQU  1Q      ;DISPLAY FUNCTIONS ENABLED
128      0002      . . .      INSHR  EQU  2Q      ;INSERT CHARACTER ENABLED
129      0004      . . .      MEMLOK EQU  4Q      ;MEMORY LOCK ENABLED
130      0008      . . .      FORMAT EQU  10Q     ;FORMAT MODE ENABLED
131      0010      . . .      EDIT   EQU  20Q     ;EDIT MODE ENABLED
132      0020      . . .      SELECT EQU  40Q     ;SELECT MODE ENABLED
133      0040      . . .      RECORD EQU  100Q    ;RECORD MODE ENABLED
134      0080      . . .      FORGN  EQU  200Q    ;FOREIGN MODE ENABLED
135      0000      . . .      ;*****
136      0000      . . .      ; MDFLG2 - TERMINAL MODE FLAGS 2 *
137      0000      . . .      ;*****
138      0001      . . .      CAPSLK EQU  1Q      ;CAPS LOCK ENABLED
139      0002      . . .      BLKMDE EQU  2Q      ;BLOCK MODE ENABLED
140      0004      . . .      AUTOLF EQU  4Q      ;AUTO LF ENABLED
141      0008      . . .      REMOTE EQU  10Q     ;REMOTE ENABLED
142      0020      . . .      WBSK   EQU  40Q     ;WRITE-BACKSPACE-READ MODE
143      0000      . . .      ;*****
144      0000      . . .      ; RADIX - BASE OF INPUT PARAMETER FOR ESC SEQ *
145      0000      . . .      ;*****
146      000A      . . .      DECRDX EQU  10      ;DECIMAL NUMBERS
147      0008      . . .      OCTRDX EQU  8       ;OCTAL NUMBERS
=====
```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	5
149	0000	.	*****		
150	0000	.	; COMMON VARIABLES *		
151	0000	.	*****		
152	9165	.	INTVEC EQU FSTRAM+1450 ;CENTRAL INTERRUPT VECTOR		
153	9168	.	SCNVEC EQU INTVEC+3 ;FOREIGN TERMINAL DISPLY SCA		
154	0000	.	;		
155	FFFF	.	COMMON EQU 1777770 ;UPPER LIMIT OF COMMON AREA		
156	00FF	.	CMBASE EQU COMMON/256 ;MSB OF COMMON ADDRESSES		
157	FF00	.	CMSTOR EQU CMBASE*256 ;MSB ADJUSTMENT FACTOR		
158	0000	.	;		
159	FFFE	.	DISPST EQU COMMON-1 ;DISPLAY REFRESH START PTR		
160	FFFD	.	TRMTYP EQU DISPST-1 ;TERMINAL TYPE NUMBER		
161	FFFC	.	KBDCSW EQU TRMTYP-1 ;KEYBOARD DATACOM SWITCHES		
162	FFFB	.	KBJMPR EQU KBDCSW-1 ;KEYBOARD STRAPS		
163	FFFA	.	KBJMP2 EQU KBJMPR-1 ;SET 2		
164	FFF9	.	KBJMP3 EQU KBJMP2-1 ;SET 3		
165	FFF8	.	CMFLGS EQU KBJMP3-1 ;COMMON FLAGS		
166	FFF7	.	ERRFLG EQU CMFLGS-1 ;ERROR FLAGS		
167	FFF6	.	INTFLG EQU ERRFLG-1 ;INTERRUPT FLAG		
168	FFF5	.	PRCCTL EQU INTFLG-1 ;PROCESSOR CONTROL FLAGS		
169	FFF4	.	MDFLG1 EQU PRCCTL-1 ;TERMINAL MODE FLAGS 1		
170	FFF3	.	MDFLG2 EQU MDFLG1-1 ;AND 2		
171	FFF1	.	MSGPT1 EQU MDFLG2-2 ;MESSAGE POINTERS		
172	FFEF	.	MSGPT2 EQU MSGPT1-2 ;.		
173	FFED	.	MSGPT3 EQU MSGPT2-2 ;.		
174	FFEB	.	MSGPT4 EQU MSGPT3-2 ;.		
175	FFE9	.	MSGPT5 EQU MSGPT4-2 ;.		
176	FFE7	.	MSGPT6 EQU MSGPT5-2 ;.		
177	FFE5	.	MSGPT7 EQU MSGPT6-2 ;.		
178	FFE3	.	MSGPT8 EQU MSGPT7-2 ;.		
179	FFE1	.	CTIVEC EQU MSGPT8-2 ;CTU INTERRUPT VECTOR		
180	FFE0	.	CTIJMP EQU CTIVEC-1 ;JUMP CODE FOR VECTOR		
181	FFDE	.	IODATA EQU CTIJMP-2 ;ESQ SEQ PARM ACCUMULATOR		
182	FFDD	.	IOCSGN EQU IODATA-1 ;SIGN FOR PARAMETER		
183	FFDC	.	IOPSGN EQU IOCSGN-1 ;PARAMETER SIGN		
184	FFDB	.	PARM1 EQU IOPSGN-1 ;ESCAPE SEQUENCE PARAMETERS		
185	FFDA	.	PARM2 EQU PARM1-1 ;.		
186	FFD9	.	PARM3 EQU PARM2-1 ;.		
187	FFD8	.	PARM4 EQU PARM3-1 ;.		
188	FFD7	.	PARM5 EQU PARM4-1 ;.		
189	FFD5	.	PARM6 EQU PARM5-2 ;.		
190	FFD4	.	RADIX EQU PARM6-1 ;RADIX OF PARAMETERS		
191	FFD2	.	RNGTA EQU RADIX-2 ;CHAR FUNCTION TABLE ADDRESS		
192	FFD1	.	ESCFLG EQU RNGTA-1 ;ESCAPE SEQUENCE FLAG		
193	0000	.	;		
194	0000	.	;		
195	FFD0	.	RSTTMR EQU ESCFLG-1 ;SOFT RESET TIMER		
196	0000	.	;*****		
197	0000	.	; END OF COMMON EQUATES		
198	0000	.	*****		

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE   6
=====
200      0000      . . .      ;*****
201      0000      . . .      ;
202      0000      . . .      ;          DATACOM CONSTANTS
203      0000      . . .      ;
204      0000      . . .      ;*****
205      5000      . . .      ZDCBAS EQU 50000          ;DATACOM START ADDRESS
206      5002      . . .      TRIGGR EQU ZDCBAS+2      ;BLOCK TRANSFER TRIGGER
207      5003      . . .      RECSEP EQU TRIGGR+1      ;RECORD SEPARATOR CHARACTER
208      5004      . . .      BLKTRM EQU RECSEP+1      ;BLOCK TERMINATOR CHARACTER
209      5005      . . .      DCJMSK EQU BLKTRM+1      ;DATA COMM JUMPER MASK
210      5006      . . .      DCJMS2 EQU DCJMSK+1      ;DATA COMM JUMPER MASK #2
211      0000      . . .      ;*****
212      0000      . . .      ;
213      0000      . . .      ;          DATACOM ENTRY VECTOR POINTERS
214      0000      . . .      ;
215      0000      . . .      ;*****
216      5008      . . .      ZINIDC EQU ZDCBAS+100     ;INITIALIZE DATACOM
217      500B      . . .      ZIN2DC EQU ZINIDC+3      ;INITIALIZATION CONTINUATUR
218      500E      . . .      ZDCMON EQU ZIN2DC+3      ;MONITORING ROUTINE
219      5011      . . .      ZDCCTL EQU ZDCMON+3      ;MISC CONTROL FUNCTIONS
220      5014      . . .      ZDCTST EQU ZDCCTL+3      ;SELF-TEST
221      5017      . . .      ZGETDC EQU ZDCTST+3      ;GET DC CHARACTER
222      501A      . . .      ZPUTDC EQU ZGETDC+3      ;PUT DC CHARACTER
223      501D      . . .      ZGTBIN EQU ZPUTDC+3      ;GET BINARY DC CHARACTER
224      5020      . . .      ZSTBIN EQU ZGTBIN+3      ;START BINARY OUTPUT
225      5023      . . .      ZNDBIN EQU ZSTBIN+3      ;END BINARY OUTPUT
226      5026      . . .      ZDCINT EQU ZNDBIN+3      ;DATACOM INTERRUPTS
227      0000      . . .      ;*****
228      0000      . . .      ;
229      0000      . . .      ;          DATACOM CONTROL CALL CODES
230      0000      . . .      ;
231      0000      . . .      ;*****
232      0000      . . .      CLRTRG EQU 0              ;CLEAR BLOCK TRANSFER TRIGGE
233      0001      . . .      SETTRG EQU 1              ;SET BLOCK TRANSFER TRIGGER
234      0002      . . .      RSETDC EQU 2              ;RESET DATACOM
235      0003      . . .      SETREM EQU 3              ;SET REMOTE MODE
236      0004      . . .      SETLCL EQU 4              ;SET LOCAL MODE
237      0005      . . .      PUTBRK EQU 5              ;OUTPUT BREAK SIGNAL
238      0006      . . .      DISCNT EQU 6              ;MODEM DISCONNECT
239      0007      . . .      ENDBLK EQU 7              ;TERMINATE OUTPUT MESSAGE
240      0008      . . .      SETMON EQU 8              ;ENTER MONITOR MODE
241      0009      . . .      SETNRM EQU 9              ;ENTER NORMAL MODE
242      000A      . . .      FSTBIN EQU 10             ;ENTER FAST BINARY OUT MODE
243      000B      . . .      SNDATN EQU 11             ;SEND ATTENTION CODE
244      000C      . . .      SDFDCT EQU 12             ;SEND FUNCTION DATA
245      000D      . . .      PROMPT EQU 13            ;SEND PROMPT CODE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
247	0000	.	.	*****	
248	0000	.	.	; MAIN CODE ENTRY VECTORS *	
249	0000	.	.	*****	
250	0040	.	.	MNCDBS EQU 100Q	
251	0040	.	.	ZDSPMS EQU MNCDBS ;DISPLAY MESSAGE	
252	0043	.	.	ZRSTDS EQU ZDSPMS+3 ;RESTORE NORMAL DISPLAY	
253	0046	.	.	ZDCNUM EQU ZRSTDS+3 ;ACCUMULATE DIGIT FOR ESC SE	
254	0049	.	.	ZDCPLS EQU ZDCNUM+3 ;ADD IN PLUS SIGN	
255	004C	.	.	ZDCMNS EQU ZDCPLS+3 ;ADD IN MINUS SIGN	
256	004F	.	.	ZESCND EQU ZDCMNS+3 ;TERMINATE ESCAPE SEQUENCE	
257	0000	.	.	*****	
258	0000	.	.	; KBFLGS - KEYBOARD ROUTINE LOCAL FLAGS *	
259	0000	.	.	*****	
260	0001	.	.	KBLOCK EQU 1Q ;KEYBOARD DISABLED	
261	0002	.	.	PERMBM EQU 2Q ;PERMANENT BLOCK MODE	
262	0008	.	.	RPTKY EQU 10Q ;REPEAT LAST KEY HIT	
263	0000	.	.	*****	
264	0000	.	.	; KBLEDS - LOCAL KEYBOARD LED EQUATES *	
265	0000	.	.	*****	
266	0001	.	.	DSFLED EQU 1Q ;DISPLAY FUNCTIONS	
267	0002	.	.	ICHLED EQU 2Q ;INSERT CHARACTER	
268	0004	.	.	MLKLED EQU 4Q ;MEMORY LOCK	
269	0008	.	.	XMTLED EQU 10Q ;TRANSMIT LED	
270	0010	.	.	EDTLED EQU 20Q ;EDIT MODE	
271	0020	.	.	SELLED EQU 40Q ;SELECT MODE	
272	0040	.	.	RECLEL EQU 100Q ;RECORD MODE	
273	0080	.	.	BELLEL EQU 200Q ;BELL	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE   8
=====
 275     0000      . . .      ;*****
 276     0000      . . .      ; MISCELLANEOUS EQUATES *
 277     0000      . . .      ;*****
 278     0006      . . .      RPTDLY EQU  6          ;REPEAT DELAY (50 MSEC)
 279     0033      . . .      SRTDLY EQU  51         ;SHORT START DELAY (500 MSEC)
 280     0058      . . .      LNGDLY EQU  91         ;LONG START DELAY (900 MSEC)
 281     0000      . . .      ;
 282     001E      . . .      BLKDLY EQU  30         ;LED BLINK DELAY (30x10 MSEC)
 283     00FF      . . .      BLKSET EQU  3770       ;BLINK SET FLAG
 284     0000      . . .      ;
 285     000D      . . .      NUMCOL EQU  13         ;# OF KEYBOARD COLUMNS - 1
 286     0003      . . .      REMBIT EQU  3          ;REMOTE FLAG BIT NUMBER
 287     0000      . . .      ;
 288     0000      . . .      ;*****
 289     0000      . . .      ; ADDED FOR GRAPHICS
 290     0097      . . .      GRAFUN EQU  2270       ;TOP OF GRAPHICS FUNCTIONS
 291     0000      . . .      ;*****
 292     0000      . . .      ;
 293     0000      . . .      ;*****
 294     0000      . . .      ; ASCII CHARACTER EQUATES *
 295     0000      . . .      ;*****
 296     0008      . . .      BKSPCE EQU  100        ;ASCII BACKSPACE CONTROL COD
 297     0009      . . .      TAB      EQU  110        ;ASCII TAB CONTROL CODE
 298     001F      . . .      CTLMSK EQU  370        ;MASK FOR CONTROL CODES
 299     0020      . . .      CPSADJ EQU  400        ;CAPS LOCK ADJUSTMENT FACTOR
 300     0028      . . .      PLUS    EQU  530        ;(+) - PLUS SIGN
 301     002F      . . .      SLANT   EQU  570        ;(/) - SLANT
 302     0030      . . .      ZERO    EQU  600        ;ASCII ZERO
 303     0020      . . .      ABLNK   EQU  400        ;ASCII BLANK
 304     0040      . . .      UPRLIM  EQU  1000       ;UPPER CASE LOWER LIMIT
 305     0041      . . .      A        EQU  1010       ;UPPER CASE A
 306     005A      . . .      Z        EQU  1320       ;UPPER CASE Z
 307     0060      . . .      LWRLIM  EQU  1400       ;LOWER CASE CHAR LOWER LIMIT
 308     007F      . . .      DEL     EQU  1770       ;DELETE CHARACTER ( = RUBOUT
 309     0000      . . .      ;
 310     8000      . . .      B15     EQU  100000      ;BIT 15 CONSTANT
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
312	0000	.	;	9
313	0000	.	; FUNCTION ESCAPE CODES	
314	0000	.	;	
315	00B2	.	CLRTAB EQU 2620 ;2 - CLEAR TAB	
316	00B3	.	CLRTBS EQU 2630 ;3 - CLEAR ALL TABS	
317	00B4	.	SETLMG EQU 2640 ;4 - SET LEFT MARGIN	
318	00B5	.	SETRMG EQU 2650 ;5 - SET RIGHT MARGIN	
319	00FF	.	ENHNCF EQU 3770 ;DISPLAY ENHANCEMENT FUNCTIO	
320	00C3	.	CURRHT EQU 3030 ;C - CURSOR RIGHT	
321	00C4	.	CURLFT EQU 3040 ;D - CURSOR LEFT	
322	00C6	.	HOMEDN EQU 3060 ;F - HOME DOWN	
323	00CA	.	CLSCRN EQU 3120 ;J - CLEAR SCREEN	
324	00CB	.	CLRLNE EQU 3130 ;K - CLEAR LINE	
325	00CE	.	IWRPON EQU 3160 ;N - INSERT CHAR W/WRAP ON	
326	00CF	.	DCHWRP EQU 3170 ;O - DELETE CHAR W/WRAPAROUN	
327	00D0	.	DELCHR EQU 3200 ;P - DELETE CHARACTER	
328	00D1	.	ICHRON EQU 3210 ;Q - INSERT CHARACTER ON	
329	00D2	.	ICHR0F EQU 3220 ;R - INSERT CHARACTER OFF	
330	00D5	.	NEXTPG EQU 3250 ;U - NEXT PAGE	
331	00D7	.	FMTONF EQU 3270 ;W - FORMAT MODE ON	
332	00D8	.	FMTOFF EQU 3300 ;X - FORMAT MODE OFF	
333	00D9	.	DSPFON EQU 3310 ;Y - DISPLAY FUNCTIONS ON	
334	00DB	.	ENDPRF EQU 3330 ;L - END PROTECTED FIELD	
335	00DD	.	STPRF EQU 3350 ;] - START PROTECTED FIELD	
336	0000	.	;	
337	00E8	.	HOMEUP EQU 3500 LOWER CASE ;H - HOME UP TO UNPROTECT	
338	00E9	.	BACKTB EQU 3510 ;I - BACK TAB	
339	00EA	.	SETSFK EQU 3520 ;J - SET SOFT KEY DEFINE ON	
340	00EC	.	MLKON EQU 3540 ;L - MEMORY LOCK ON	
341	00F0	.	F1FUNC EQU 3600 ;P - F1 FUNCTION CODE	
342	00F9	.	MNMDON EQU 3710 ;Y - MONITOR MODE ON	
343	00FA	.	TEST EQU 3720 ;Z - SELF-TEST	
344	00FB	.	STXM0F EQU 3730 ;[- START TRANSMIT ONLY	
345	0000	.	;	
346	0000	.	; EXTERNAL FUNCTION CODES	
347	0000	.	;	
348	0098	.	ENTRCD EQU 2300 ;ENTER KEY	
349	009A	.	DFNCOF EQU 2320 ;DISPLAY FUNCTIONS OFF	
350	009C	.	RDKYCD EQU 2340 ;READ KEY	
351	009F	.	CONDTN EQU 2370 ;CONDITION TAPE	
352	00A0	.	CTRDKY EQU 2400 ;CONTROL READ KEY	
353	00A1	.	EXFNLM EQU 2410 ;EXTERNAL FUNCTION UPPER LIM	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 10
=====
355      0000      . . .      ;
356      0000      . . .      ; KEYBOARD SLOW RAM EQUATES
357      0000      . . .      ;
358      FF20      . . .      KBSRAM EQU 1774400 ;UPPER LIMIT OF SLOW RAM ARE
359      00FF      . . .      KBSBSE EQU KBSRAM-1/256 ;MSB OF SLOW RAM AREA
360      FF00      . . .      KBSSTR EQU KBSBSE*256 ;MSB ADJUSTMENT FACTOR
361      FF12      . . .      KBBUF2 EQU KBSRAM-14 ;ACTIVE KEYS STATE TABLE
362      FF11      . . .      KBFLGS EQU KBBUF2-1 ;KEYBOARD FLAGS
363      FF10      . . .      KBJ1MS EQU KBFLGS-1 ;JUMPER ALTER INHIBIT MASK -
364      0000      . . .      ; SET TO 1 IF NOT TO BE
365      0000      . . .      ; ALTERED BY ESCAPE SEQUENCE
366      FF0F      . . .      KBCHAR EQU KBJ1MS-1 ;REPEATING KEY CHARACTER
367      FF0E      . . .      BLKFLG EQU KBCHAR-1 ;LED BLINK FLAG
368      FF0D      . . .      KBKNSV EQU BLKFLG-1 ;REPEATING KEY NUMBER
369      FF0C      . . .      KBLEDS EQU KBKNSV-1 ;STATE OF KEYBOARD LED'S
370      FF0B      . . .      LEDSAV EQU KBLEDS-1 ;SAVE AREA FOR LED VALUES
371      0000      . . .      ;
372      0000      . . .      ; KEYBOARD FAST ACCESS RAM EQUATES
373      0000      . . .      ;
374      9200      . . .      KBFRAM EQU FSTRAM+4000 ;UPPER LIMIT OF FAST RAM
375      0091      . . .      KBFBSE EQU KBFRAM-1/256 ;MSB OF FAST RAM AREA
376      9100      . . .      KBFSTR EQU KBFBSE*256 ;MSB ADJUSTMENT FACTOR
377      0000      . . .      ;
378      91F2      . . .      KBBUF EQU KBFRAM-14 ;COLUMN STATE TABLE BUFFER
379      00F2      . . .      KBBUFL EQU KBBUF-KBFSTR ;LSB OF STATE TABLE ADDR
380      91F0      . . .      KBBFPT EQU KBBUF-2 ;KEYBOARD STATE TABLE POINTE
381      91EE      . . .      KEYCOL EQU KBBFPT-2 ;KEYBOARD COLUMN ADDRESS
382      91ED      . . .      BLKTMR EQU KEYCOL-1 ;BLINK DELAY TIMER
383      91EC      . . .      KBTIMR EQU BLKTMR-1 ;KEY REPEAT TIMER
384      0028      . . .      KEYBLN EQU 40 ;KEY BUFFER LENGTH
385      91C4      . . .      KEYBUF EQU KBTIMR-KEYBLN ;KEY BUFFER
386      0000      . . .      ; EACH TWO BYTE ENTRY IN THE
387      0000      . . .      ; KEY BUFFER REPRESENTS ONE
388      0000      . . .      ; OR MORE KEY TRANSITIONS.
389      0000      . . .      ; THE MSB IS THE KEYBOARD
390      0000      . . .      ; COLUMN NUMBER AND THE LSB,
391      0000      . . .      ; THE BITS CHANGED.
392      00C4      . . .      KEYBFL EQU KEYBUF-KBFSTR ;LSB OF KEY BUFFER ADDR
393      91C2      . . .      KBPTPT EQU KEYBUF-2 ;KEY BUFFER PUT POINTER
394      91C0      . . .      KBGTPT EQU KBPTPT-2 ;KEY BUFFER GET POINTER
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 11
=====
396      0000      . . .      ;
397      0000      . . .      ; EQUATES TO SPECIFIC COLUMNS IN STATE TABLE
398      0000      . . .      ;
399      FF12      . . .      KBCTSH EQU  KBBUF2      ;CONTROL AND SHIFT KEYS
400      0001      . . .      CTLKEY EQU  1Q          ;CONTROL KEY BIT
401      0008      . . .      LSHFKY EQU  10Q         ;LEFT SHIFT KEY
402      0010      . . .      RSHFKY EQU  20Q         ;RIGHT SHIFT KEY
403      0000      . . .      ;
404      FF17      . . .      TSTROW EQU  KBBUF2+5    ;TEST KEY ROW
405      0001      . . .      TSTCOL EQU  1Q          ;TEST KEY BIT
406      0000      . . .      ;
407      FF1B      . . .      IOKYRW EQU  KBBUF2+9    ;I/O CONTROL KEY ROW
408      0001      . . .      IOKYCL EQU  1Q          ;I/O CONTROL (GOLD) KEY
409      0002      . . .      ENTCOL EQU  2Q          ;ENTER KEY
410      0000      . . .      ;
411      0000      . . .      BRKYRN EQU  13          ;BREAK KEY ROW NUMBER
412      0001      . . .      BRKCOL EQU  1Q          ;BREAK KEY
413      0000      . . .      ;
414      0000      . . .      ; EQUATES TO SPECIFIC KEY NUMBERS
415      0000      . . .      ;
416      0008      . . .      RMKYNM EQU  10Q         ;REMOTE
417      0010      . . .      CLKYNM EQU  20Q         ;CAPS LOCK
418      0020      . . .      ALKYNM EQU  40Q         ;AUTO LINE FEED
419      0040      . . .      BMKYNM EQU  100Q        ;BLOCK MODE
=====
  
```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 12
=====
421      0000      . . .      ;*****
422      0000      . . .      ; I/O MODULE EQUATES *
423      0000      . . .      ;*****
424      0080      . . .      IOBASE EQU 2000      ;I/O ADDRESS MSB'S
425      8300      . . .      IOKB EQU 30+IOBASE*256 ;MODULE 11 BASE ADDRESS
426      830E      . . .      IOKBSW EQU IOKB+160 ;KEYBOARD JUMPERS IN
427      830F      . . .      IOKBDC EQU IOKB+170 ;DATACOM SWITCHES IN
428      8380      . . .      IOKBS2 EQU IOKB+2000 ;KEYBOARD JUMPERS 2 IN
429      83A0      . . .      IOKBS3 EQU IOKB+2400 ;KEYBOARD JUMPERS 3 IN
430      0000      . . .      ;
431      8300      . . .      IOKBLD EQU IOKB+00 ;SET KEYBOARD LED'S
432      8380      . . .      IOKBCG EQU IOKB+2000 ;RESET KEY CONTROL
433      0002      . . .      RSTON EQU 20 ;ENABLE RESET KEY
434      0004      . . .      RSTOFF EQU 40 ;DISABLE RESET KEY
435      8320      . . .      IOKBCL EQU IOKB+400 ;OUTPUT LAST COLUMN STATE
=====
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
437	0000	.	.	;*****	13
438	0000	.	.	; ENTRY VECTORS TO GRAPHICS ROUTINES	
439	0000	.	.	;*****	
440	600E	.	.	ZGCKYS EQU 60016Q ;CHECK FOR CURSOR KEY PRESSE	
441	6011	.	.	ZRELGC EQU 60021Q ;CHK FOR CURSOR KEY RELEASED	
442	6014	.	.	ZTINT EQU 60024Q ;TIMER INTERRUPT	
443	603E	.	.	ZTKCLR EQU 60076Q ;CLEAR ECHO SUPRESS	
444	00C6	.	.	ZCHKSF EQU 306Q ;CHECK FOR SOFT KEYS UP	
445	6023	.	.	ZMUCHK EQU 60043Q ;CHECK FOR AP MENU UP	
446	00EF	.	.	SFTCR EQU 357Q ;SOFT RETURN KEY CODE	
447	000D	.	.	CR EQU 15Q ;HARD RETURN	
448	6059	.	.	ZTKSTR EQU 60131Q ;SET TEK STRAPS	
449	008B	.	.	ZOOMIN EQU 213Q ;ZOOM IN KEYCODE	
450	008C	.	.	ZMOUT EQU 214Q ;ZOOM OUT KEYCODE	
451	0010	.	.	ZMDLY EQU 20Q ;LONGER DELAY FOR ZOOM	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 14
=====
453      0000      . . .      ;
454      0000      . . .      ;*****
455      0000      . . .      ; START OF KEYBOARD CODE *
456      0000      . . .      ;*****
457      0000      . . .      ORG 44000Q      ;START OF KEYBOARD CODE
458      4800      . . .      KBBASE EQU $      ;REGION (2K)
459      4800      . . .      ;*****
460      4800      54 . .      DB 124Q      ;GRAPHICS VERSION = 'T'
461      4801      . . .      ;*****
462      4801      48 . .      DB KBBASE/256
463      4802      . . .      ;
464      4802      . . .      ; KEYBOARD ENTRY VECTOR
465      4802      . . .      ;
466      4802      C3 43 49      JMP INITKB      ;KEYBOARD INITIALIZATION
467      4805      C3 75 49      JMP GTKEY      ;GET KEYBOARD KEY
468      4808      C3 52 4C      JMP KBCTL      ;PERFORM KEYBOARD CONTROL
469      480B      C3 99 4B      JMP KBMUN      ;MONITOR KEYBOARD
470      480E      C3 78 4D      JMP SETMD1      ;SET MODE 1 FLAGS
471      4811      C3 9F 4D      JMP CLRMD1      ;CLEAR MODE 1 FLAGS
472      4814      C3 D0 4D      JMP BELL      ;SOUND THE KEYBOARD BELL
473      4817      C3 DE 4D      JMP SETxMT      ;SET TRANSMIT LED
474      481A      C3 E7 4D      JMP CLRxMT      ;CLEAR TRANSMIT LED
475      481D      C3 F2 4D      JMP STJMPR      ;SET JUMPER ESC SEQ ROUTINE
476      4820      C3 75 4E      JMP STLKYS      ;SET LATCHING KEYS ESC SEQ
477      4823      C3 D4 4E      JMP ALPCHK      ;CHECK FOR ALPHA KEY ENTRY
478      4826      C3 E1 4E      JMP NUMCHK      ;CHECK FOR NUMERIC KEY ENTRY
479      4829      . . .      ;
480      4829      . . .      ; KEYBOARD CONSTANTS
481      4829      . . .      ;
482      4829      10 . .      FRSAIT DB 20Q      ;INITIAL ALT CHAR SET = SET
483      482A      00 . .      ALTOUT DB 0Q      ;INITIAL ALT CHAR SET OUT =
484      482B      . . .      ;*****
485      482B      . . .      ; ADDED FOR GRAPHICS
486      482B      00 . .      DB 0      ;I/O CODE EXPECTS 0 HERE
487      482C      37 48 .      DW LWRASC      ;ADDRESS OF LOWER CASE TABLE
488      482E      C3 FB 4E      JMP RETSCN      ;'RETSCN' ROUTINE MOVED FROM
489      4831      C3 F1 4E      JMP RETSCU      ;I/O CODE
490      4834      C3 16 4F      JMP USRINT
491      4837      . . .      ;*****

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 15
=====
493      4837      . . .      ;
494      4837      . . .      ; LOWER CASE ASCII TRANSLATION TABLE
495      4837      . . .      ;
496      4837      . . .      ;*****
497      4837      . . .      ; MODIFIED FOR GRAPHICS
498      4837      . . .      ;*****
499      4837      . . .      LWRASC EQU $
500      4837      00 1B 09      DB 000Q,033Q,011Q,000Q ;COLUMN 0
501      483B      00 8B A4      DB 000Q,213Q,244Q,010Q
502      483F      86 31 71      DB 206Q,061Q,161Q,172Q ;COLUMN 1
503      4843      EF A3 87      DB 357Q,243Q,207Q,134Q
504      4847      83 32 77      DB 203Q,062Q,167Q,170Q ;COLUMN 2
505      484B      5D 8C A2      DB 135Q,214Q,242Q,364Q
506      484F      80 33 65      DB 200Q,063Q,145Q,143Q ;COLUMN 3
507      4853      3A C4 D3      DB 072Q,304Q,323Q,365Q
508      4857      84 34 72      DB 204Q,064Q,162Q,166Q ;COLUMN 4
509      485B      3B E8 C1      DB 073Q,350Q,301Q,366Q
510      485F      FA 35 74      DB 372Q,065Q,164Q,142Q ;COLUMN 5
511      4863      6C C3 D5      DB 154Q,303Q,325Q,367Q
512      4867      C5 36 79      DB 305Q,066Q,171Q,040Q ;COLUMN 6
513      486B      6B D6 CA      DB 153Q,326Q,312Q,202Q
514      486F      81 37 75      DB 201Q,067Q,165Q,156Q ;COLUMN 7
515      4873      6A C2 B1      DB 152Q,302Q,261Q,320Q
516      4877      85 38 69      DB 205Q,070Q,151Q,155Q ;COLUMN 8
517      487B      68 D4 B2      DB 150Q,324Q,262Q,315Q
518      487F      9B 98 6F      DB 233Q,230Q,157Q,054Q ;COLUMN 9
519      4883      67 8A 88      DB 147Q,212Q,210Q,314Q
520      4887      9C 39 70      DB 234Q,071Q,160Q,056Q ;COLUMN 10
521      488B      66 A5 A1      DB 146Q,245Q,241Q,363Q
522      488F      9D 30 40      DB 235Q,060Q,100Q,057Q ;COLUMN 11
523      4893      64 00 93      DB 144Q,000Q,223Q,362Q
524      4897      9E 2D 5B      DB 236Q,055Q,133Q,000Q ;COLUMN 12
525      489B      73 00 00      DB 163Q,000Q,000Q,361Q
526      489F      99 5E 5F      DB 231Q,136Q,137Q,000Q ;COLUMN 13
527      48A3      61 00 00      DB 141Q,000Q,000Q,360Q
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 16
=====
529      48A7      . . .      ;
530      48A7      . . .      ; UPPER CASE ASCII TABLE
531      48A7      . . .      ;
532      48A7      . . .      ;*****
533      48A7      . . .      ; MODIFIED FOR GRAPHICS
534      48A7      . . .      ;*****
535      48A7      . . .      UPASC EQU $
536      48A7      00 18 09      DB 000Q,033Q,011Q,000Q ;COLUMN 0
537      48AB      00 8D 90      DB 000Q,215Q,220Q,010Q
538      48AF      86 21 51      DB 206Q,041Q,121Q,132Q ;COLUMN 1
539      48B3      EF 96 8F      DB 357Q,226Q,217Q,174Q
540      48B7      83 22 57      DB 203Q,042Q,127Q,130Q ;COLUMN 2
541      48BB      7D 97 91      DB 175Q,227Q,221Q,364Q
542      48BF      80 23 45      DB 200Q,043Q,105Q,103Q ;COLUMN 3
543      48C3      2A C4 D3      DB 052Q,304Q,323Q,365Q
544      48C7      84 24 52      DB 204Q,044Q,122Q,126Q ;COLUMN 4
545      48CB      2B E8 C1      DB 053Q,350Q,301Q,366Q
546      48CF      FA 25 54      DB 372Q,045Q,124Q,102Q ;COLUMN 5
547      48D3      4C C3 D5      DB 114Q,303Q,325Q,367Q
548      48D7      C5 26 59      DB 305Q,046Q,131Q,040Q ;COLUMN 6
549      48DB      4B D6 CA      DB 113Q,326Q,312Q,202Q
550      48DF      81 27 55      DB 201Q,047Q,125Q,116Q ;COLUMN 7
551      48E3      4A C2 B1      DB 112Q,302Q,261Q,320Q
552      48E7      85 28 49      DB 205Q,050Q,111Q,115Q ;COLUMN 8
553      48EB      48 D4 B2      DB 110Q,324Q,262Q,315Q
554      48EF      9B 98 4F      DB 233Q,230Q,117Q,074Q ;COLUMN 9
555      48F3      47 95 8E      DB 107Q,225Q,216Q,314Q
556      48F7      9C 29 50      DB 234Q,051Q,120Q,076Q ;COLUMN 10
557      48FB      46 92 89      DB 106Q,222Q,211Q,363Q
558      48FF      9D 30 60      DB 235Q,060Q,140Q,077Q ;COLUMN 11
559      4903      44 00 94      DB 104Q,000Q,224Q,362Q
560      4907      9E 3D 7B      DB 236Q,075Q,173Q,000Q ;COLUMN 12
561      490B      53 00 00      DB 123Q,000Q,000Q,361Q
562      490F      99 7E 7F      DB 231Q,176Q,177Q,000Q ;COLUMN 13
563      4913      41 00 00      DB 101Q,000Q,000Q,360Q
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
565	4917	.	.	;*****	17
566	4917	.	.	; ALTERNATE FUNCTION KEY FUNCTIONS TABLE *	
567	4917	.	.	;*****	
568	4917	.	.	;	
569	4917	.	.	; NORMAL FUNCTION CODES FOLLOWED BY ALTERNATES	
570	4917	.	.	;	
571	4917	.	.	NRMFCT EQU \$	
572	4917	9C	.	DB RDKYCD ;READ KEY	
573	4918	A0	.	DB CTRDKY ;CONTROL READ KEY	
574	4919	B2	.	DB CLR TAB ;CLEAR TAB	
575	491A	B3	.	DB CLRTBS ;CLEAR ALL TABS	
576	491B	C4	.	DB CURLFT ;CURSOR LEFT	
577	491C	B4	.	DB SETLMG ;SET LEFT MARGIN	
578	491D	C3	.	DB CURRHT ;CURSOR RIGHT	
579	491E	B5	.	DB SETRMG ;SET RIGHT MARGIN	
580	491F	CA	.	DB CLSCRN ;CLEAR SCREEN	
581	4920	CB	.	DB CLRLNE ;CLEAR LINE	
582	4921	D0	.	DB DELCHR ;DELETE CHARACTER	
583	4922	01	.	DB CHECK1 ;ADDITIONAL CHECK 1	
584	4923	D5	.	DB NEXTPG ;NEXT PAGE	
585	4924	02	.	DB CHECK2 ;ADDITIONAL CHECK 2	
586	4925	E8	.	DB HOMEUP ;HOME UP	
587	4926	C6	.	DB HOMEDN ;HOME DOWN	
588	4927	FA	.	DB TEST ;SELF-TEST	
589	4928	9F	.	DB CONDTN ;CONDITION TAPE	
590	4929	F0	.	DB F1FUNC ;F1 KEY	
591	492A	FF	.	DB ENHNCF ;START DISPLAY ENHANCEMENT	
592	492B	F1	.	DB F1FUNC+1 ;F2 KEY	
593	492C	DB	.	DB ENDPFR ;END PROTECTED FIELD	
594	492D	F2	.	DB F1FUNC+2 ;F3 KEY	
595	492E	DD	.	DB STPRF ;START PROTECTED FIELD	
596	492F	F3	.	DB F1FUNC+3 ;F4 KEY	
597	4930	D7	.	DB FMTONF ;FORMAT MODE ON	
598	4931	F4	.	DB F1FUNC+4 ;F5 KEY	
599	4932	D8	.	DB FMTOFF ;FORMAT MODE OFF	
600	4933	F5	.	DB F1FUNC+5 ;F6 KEY	
601	4934	FB	.	DB STXM OF ;START TRANSMIT-ONLY FIELD	
602	4935	.	.	;	
603	000F	.	.	NUMALT EQU \$-NRMFCT/2 ;# OF ALTERNATE FUNCTIONS	
604	0001	.	.	CHECK1 EQU 1 ;EXTRA CHECK 1 FLAG	
605	0002	.	.	CHECK2 EQU 2 ;EXTRA CHECK 2 FLAG	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  18
=====
607      4935      . . .      ;
608      4935      . . .      ; INTERNAL FUNCTIONS VECTOR TABLE
609      4935      . . .      ;
610      4935      . . .      FNCTAB EQU $
611      4935      2E 4B .      DW  CKMLOK    ;200 - MEMORY LOCK
612      4937      3C 4B .      DW  CKDSFN    ;201 - DISPLAY FUNCTIONS
613      4939      58 4B .      DW  CKICHR    ;202 - INSERT CHARACTER
614      493B      7C 4B .      DW  STCPLK    ;203 - CAPS LOCK
615      493D      81 4B .      DW  STAULF    ;204 - AUTO LF
616      493F      86 4B .      DW  STBLKM    ;205 - BLOCK MODE
617      4941      8B 4B .      DW  STRMMD    ;206 - SET REMOTE MODE
618      4943      . . .      ;
619      0080      . . .      FNBASE EQU 2000 ;FUNCTION LOWER LIMIT
620      0087      . . .      FNCLIM EQU 2070 ;FUNCTION CODE UPPER LIMIT
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
622	4943	.	.	;	19
623	4943	.	.	; * * * * *	
624	4943	.	.	;	
625	4943	.	.	; INITKB - INITIALIZE KEYBOARD	
626	4943	.	.	;	
627	4943	.	.	; ENTRY DON'T CARE	
628	4943	.	.	;	
629	4943	.	.	; EXIT A DESTROYED	
630	4943	.	.	; NC = NO ERRORS	
631	4943	.	.	; C = ERROR DETECTED	
632	4943	.	.	; B,C = POINTER TO ERROR MESSAGE	
633	4943	.	.	;	
634	4943	.	.	INITKB EQU \$	
635	4943	3A	0E 83	LDA IOKBSW ;GET KEYBOARD STRAP SETTINGS	
636	4946	32	FB FF	STA KBJMPR ;AND STORE THEM	
637	4949	3A	80 83	LDA IOKBS2	
638	494C	32	FA FF	STA KBJMP2	
639	494F	3A	A0 83	LDA IOKBS3	
640	4952	32	F9 FF	STA KBJMP3	
641	4955	3A	0F 83	LDA IOKBDC ;GET AND STORE THE DATA	
642	4958	32	FC FF	STA KBDCSW ;COMM SWITCHES	
643	495B	.	.	;	
644	495B	.	.	INITK0 EQU \$	
645	495B	21	FF 91	LXI H,KBBUF+NUMCOL	
646	495E	22	F0 91	SHLD KBBFPT ;INITIALIZE STATE TABLE	
647	4961	21	0D 83	LXI H,IOKB+NUMCOL ;POINTERS	
648	4964	22	EE 91	SHLD KEYCOL	
649	4967	21	EB 91	LXI H,KEYBUF+KEYBLN-1	
650	496A	22	C0 91	SHLD KBGTPT ;INITIALIZE TRANSITION	
651	496D	22	C2 91	SHLD KBPTPT ;BUFFER POINTERS	
652	4970	.	.	;	
653	4970	AF	.	XRA A ;CLEAR THE LEDS ON THE	
654	4971	.	.	INITK1 EQU \$;KEYBOARD	
655	4971	32	00 83	STA IOKBLD	
656	4974	C9	.	RET ;RETURN C = FALSE	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 20
658	4975	. . .	;*****	
659	4975	. . .	; GTKEY - GET A KEYBOARD CHARACTER *	
660	4975	. . .	;*****	
661	4975	. . .	; ENTRY DON'T CARE	
662	4975	. . .	; EXIT Z - KEYBOARD INPUT PRESENT	
663	4975	. . .	; A = KEYBOARD CHARACTER CODE	
664	4975	. . .	; NZ - NO KEYBOARD INPUT	
665	4975	. . .	; A = 0, NO KEY HIT (OR NULL CHAR)	
666	4975	. . .	; A # 0, KEYBOARD LOCKED (A IS	
667	4975	. . .	; KEYBOARD CHARACTER CODE)	
668	4975	. . .	; B,C,D,E DESTROYED	
669	4975	. . .	; KEY VALUES IN THE RANGE 260-376 (OCTAL)	
670	4975	. . .	; REPRESENT TWO CHARACTER ESCAPE SEQUENCES	
671	4975	. . .	; GENERATED FROM FUNCTION KEYS. THE SECOND	
672	4975	. . .	; CHARACTER IS OBTAINED BY MASKING OUT THE	
673	4975	. . .	; HIGH ORDER BIT.	
674	4975	. . .	; KEY VALUES IN THE RANGE 230-241 (OCTAL)	
675	4975	. . .	; REPRESENT INTERNAL FUNCTION KEYS AS	
676	4975	. . .	; FOLLOWS	
677	4975	. . .	; 230 - ENTER KEY PRESSED	
678	4975	. . .	; 231 - BREAK KEY PRESSED	
679	4975	. . .	; 232 - DISPLAY FUNCTIONS OFF	
680	4975	. . .	; 233 - I/O CONTROL KEY PRESSED	
681	4975	. . .	; 234 - READ KEY PRESSED	
682	4975	. . .	; 235 - RECORD KEY PRESSED	
683	4975	. . .	; 236 - SELECT KEY PRESSED	
684	4975	. . .	; 237 - CONDITION TAPE FUNCTION	
685	4975	. . .	; GTKEY EQU \$	
686	4975	E5 . .	PUSH H ;SAVE H AND L	
687	4976	CD CD 4C	CALL STRTS1 ;ENABLE RESET KEY	
688	4979	3A 11 FF	LDA KBFLGS ;GET KEYBOARD FLAGS	
689	497C	E6 08 .	ANI RPTKY ;REPEAT LAST KEY HIT?	
690	497E	C2 BE 49	JNZ GTK040 ;YES - RE-ISSUE THE KEY	
691	4981	2A C0 91	LHLD KBGTPT ;NO - FETCH BUFFER GET PTR	
692	4984	. . .	GTK005 EQU \$	
693	4984	3A C2 91	LDA KBPTPT ;GET LSB OF PUT POINTER	
694	4987	BD . .	CMP L ;BUFFER EMPTY?	
695	4988	C2 CB 49	JNZ GTK100 ;NO - PROCESS KEYBOARD INPUT	
696	498B	CD BE 4B	CALL KBMON1 ;YES - LOOK FOR STATE CHANGE	
697	498E	. . .	GTK010 EQU \$	
698	498E	2A C0 91	LHLD KBGTPT ;GET KEY BUFFER POINTERS	
699	4991	3A C2 91	LDA KBPTPT	
700	4994	BD . .	CMP L ;BUFFER EMPTY?	
701	4995	C2 CB 49	JNZ GTK100 ;NO - PROCESS KEYBOARD INPUT	


```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  21
=====
  708     4998      .   .   .   GTK020 EQU $
  709     4998     3A  0F  83   LDA  IOKBDC      ;YES - UPDATE SETTINGS OF
  710     4998     32  FC  FF   STA  KBDCSW      ;KEYBOARD DATACOM SWITCHES
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  22
=====
712      499E      . . .      ;*****
713      499E      . . .      ; CHECK FOR REPEATING KEY *
714      499E      . . .      ;*****
715      499E      21 EC 91      LXI  H,KBTIMR  ;GET THE REPEAT DELAY TIMER
716      49A1      7E . .      MOV  A,M
717      49A2      3D . .      DCR  A          ;TIME TO REPEAT?
718      49A3      C2 C7 49      JNZ  GTK060    ;NO - EXIT NO KEY HIT
719      49A6      . . .      ;*****
720      49A6      . . .      ; IF KEY IS ZOOM IN OR OUT, USE SLOWER DELAY
721      49A6      1E 06 .      MVI  E,RPTDLY ;ASSUME NORMAL DELAY
722      49A8      3A 0F FF      LDA  KBCHAR    ;FETCH THE CHAR
723      49A8      FE 88 .      CPI  ZOOMIN    ;ZOOM KEY?
724      49AD      DA B7 49      JC   GTK025    ;NO
725      49B0      FE 8D .      CPI  ZMOUT+1   ;ZOOM KEY?
726      49B2      D2 B7 49      JNC  GTK025    ;NO
727      49B5      1E 10 .      MVI  E,ZMDLY  ;YES, USE SLOWER DELAY
728      49B7      . . .      GTK025 EQU $
729      49B7      73 . .      MOV  M,E       ;STORE REPEAT DELAY
730      49B8      . . .      ;*****
731      49B8      . . .      GTK030 EQU $
732      49B8      3A 0F FF      LDA  KBCHAR    ;RECALL THE KEYBOARD CHAR
733      49B8      C3 AC 4A      JMP  GTKYX1    ;RETURN KEY HIT
734      49BE      . . .      ;*****
735      49BE      . . .      ; REPEAT LAST KEY HIT *
736      49BE      . . .      ;*****
737      49BE      . . .      GTK040 EQU $
738      49BE      21 11 FF      LXI  H,KBFLGS ;CLEAR REPEAT FLAG
739      49C1      2F . .      CMA                      ;CONVERT BIT TO CLEAR MASK
740      49C2      A6 . .      ANA  M          ;CLEAR REPEAT KEY FLAG
741      49C3      77 . .      MOV  M,A       ;UPDATE FLAGS
742      49C4      C3 B8 49      JMP  GTK030    ;RE-ISSUE THE KEY CODE
743      49C7      . . .      ;*****
744      49C7      . . .      ; NO KEY CHANGES - EXIT *
745      49C7      . . .      ;*****
746      49C7      . . .      GTK060 EQU $
747      49C7      F6 FF .      ORI  377G      ;SET NC, NZ
748      49C9      E1 . .      POP  H         ;RESTORE H,L
749      49CA      C9 . .      RET
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 23
751	49CB	.	.	;	
752	49CB	.	.	; KEYBOARD STATE CHANGE IN BUFFER	
753	49CB	.	.	;	
754	49CB	.	.	GTK100 EQU \$	
755	49CB	16	00	MVI D,0 ;SET D FOR MSB = 0	
756	49CD	5E	.	MOV E,M ;GET KEYBOARD COLUMN NUMBER	
757	49CE	2B	.	DCX H ;(D = 0)	
758	49CF	4E	.	MOV C,M ;GET NEW STATE VALUE	
759	49D0	E5	.	PUSH H ;SAVE BUFFER ADDRESS	
760	49D1	21	12 FF	LXI H,K8BUF2 ;COMPUTE LOCATION OF	
761	49D4	19	.	DAD D ;STATE TABLE BYTE	
762	49D5	7E	.	MOV A,M	
763	49D6	A9	.	XRA C ;ANY MORE CHANGES?	
764	49D7	C2	EA 49	JNZ GTK130 ;YES - GO PROCESS CHANGES	
765	49DA	E1	.	POP H ;NO - ADVANCE TO NEXT	
766	49D8	3E	C4	MVI A,KEYBFL ;BUFFER ENTRY	
767	49DD	BD	.	CMP L ;REACHED START OF BUFFER?	
768	49DE	C2	E3 49	JNZ GTK120 ;NO - MOVE TO NEXT BYTE	
769	49E1	2E	EC	MVI L,KEYBFL+KEYBLN ;YES - RESET POINTER	
770	49E3	.	.	GTK120 EQU \$	
771	49E3	2B	.	DCX H ;MOVE TO NEXT BYTE	
772	49E4	22	C0 91	SHLD K8GTPT ;UPDATE GET POINTER	
773	49E7	C3	84 49	JMP GTK005 ;CHECK FOR MORE CHANGES	
774	49EA	.	.	*****	
775	49EA	.	.	; KEYBOARD CHANGE FOUND - DETERMINE NEW STATE *	
776	49EA	.	.	*****	
777	49EA	.	.	GTK130 EQU \$	
778	49EA	33	.	INX SP ;POP OFF TOP OF STACK	
779	49EB	33	.	INX SP	
780	49EC	4F	.	MOV C,A ;EXTRACT RIGHTMOST CHANGED	
781	49ED	AF	.	XRA A ;BIT	
782	49EE	91	.	SUB C	
783	49EF	A1	.	ANA C	
784	49F0	47	.	MOV B,A ;SAVE CHANGED BIT IN B-REG	
785	49F1	AE	.	XRA M ;COMPUTE NEW STATE AND	
786	49F2	77	.	MOV M,A ;UPDATE STATE TABLE	
787	49F3	A0	.	ANA B ;WAS KEY RELEASED?	
788	49F4	C2	47 4A	JNZ GTK200 ;NO - EXTRACT KEY CODE	
789	49F7	CD	1C 4C	CALL GTKYNM ;YES - COMPUTE KEY NUMBER	
790	49FA	.	.	*****	
791	49FA	.	.	; MODIFICATION FOR GRAPHICS	
792	49FA	.	.	; IF A GRAPHICS CURSOR KEY HAS BEEN RELEASED (INDI	
793	49FA	.	.	; CATED BY THE CY FLAG) DONT PROCESS ANY FURTHER.	
794	49FA	CD	11 60	CALL ZRELGC ;G-CURSOR KEY RELEASED?	
795	49FD	DA	C7 49	JC GTK060 ;YES--STOP PROCESSING	
796	4A00	.	.	*****	
797	4A00	BE	.	CMP M ;WAS THE KEY REPEATING?	
798	4A01	C2	0B 4A	JNZ GTK150 ;NO - CHECK FOR LATCH RELEAS	
799	4A04	7A	.	MOV A,D ;CLEAR THE REPEAT TIMER	
800	4A05	32	EC 91	STA K8TIMR	

13255

13255/90010

2648A MICROCODE LISTING 'KG14'

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                PAGE  24
=====
  801     4A08     C3  8E  49                JMP  GTK010      ;TRY FOR ANOTHER KEY
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 25
803	4A0B	.	.	.	;
804	4A0B	.	.	.	;
805	4A0B	.	.	.	;
806	4A0B	.	.	.	;
807	4A0B	.	.	.	;
808	4A0B	FE	40	.	GTK150 EQU \$
809	4A0D	CA	2A	4A	CPI BMKYNM ;BLOCK MODE KEY?
810	4A10	F2	8E	49	JZ GTK160 ;YES - RESET BLOCK MODE
811	4A13	FE	20	.	JP GTK010 ;NOT LATCHING KEY - TRY AGAI
812	4A15	CA	37	4A	CPI ALKYNM ;AUTO LF KEY?
813	4A18	FE	10	.	JZ GTK170 ;YES - RESET AUTO LF FLAG
814	4A1A	CA	3C	4A	CPI CLKYNM ;CAPS LOCK KEY
815	4A1D	D6	08	.	JZ GTK180 ;YES - RESET CAPS LOCK FLAG
816	4A1F	C2	8E	49	SUI RMKYNM ;REMOTE KEY?
817	4A22	32	EC	91	JNZ GTK010 ;NO - TRY FOR ANOTHER KEY
818	4A25	3E	08	.	STA KBTIMR ;YES - CLEAR REPEAT TIMER
819	4A27	C3	3E	4A	MVI A,REMOTE ;AND CLEAR REMOTE FLAG
820	4A2A	.	.	.	JMP GTK190
821	4A2A	.	.	.	;
822	4A2A	.	.	.	;
823	4A2A	.	.	.	;
824	4A2A	3A	11	FF	GTK160 EQU \$
825	4A2D	E6	02	.	LDA KBFLGS ;GET KEYBOARD FLAGS
826	4A2F	C2	8E	49	ANI PERMBM ;SET FOR PERMANENT BLOCK MDE
827	4A32	3E	02	.	JNZ GTK010 ;YES - TRY FOR ANOTHER KEY
828	4A34	C3	3E	4A	MVI A,BLKMDE ;NO - CLEAR BLOCK MODE FLAG
829	4A37	.	.	.	JMP GTK190
830	4A37	.	.	.	;
831	4A37	.	.	.	;
832	4A37	.	.	.	;
833	4A37	3E	04	.	GTK170 EQU \$
834	4A39	C3	3E	4A	MVI A,AUTOLF ;SET FLAG BIT TO BE CLEARED
835	4A3C	.	.	.	JMP GTK190 ;CLEAR THE FLAG AND RETURN
836	4A3C	.	.	.	;
837	4A3C	.	.	.	;
838	4A3C	.	.	.	;
839	4A3C	3E	01	.	GTK180 EQU \$
840	4A3E	.	.	.	MVI A,CAPSLK ;SET FLAG BIT TO BE CLEARED
841	4A3E	.	.	.	;
842	4A3E	2F	.	.	GTK190 EQU \$
843	4A3F	21	F3	FF	CMA ;CONVERT TO CLEAR MASK
844	4A42	A6	.	.	LXI H,MDFLG2 ;CLEAR FLAG FROM TERMINAL
845	4A43	77	.	.	ANA M ;MODE FLAGS 2
846	4A44	C3	8E	49	MOV M,A ;UPDATE FLAGS
					JMP GTK010 ;TRY FOR ANOTHER KEY

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 26
=====
 848      4A47      . . .      ;
 849      4A47      . . .      ; NEW KEY IS PRESSED DOWN - COMPUTE KEY CODE
 850      4A47      . . .      ;
 851      4A47      . . .      GTK200 EQU $
 852      4A47      . . .      ;*****
 853      4A47      . . .      ; MODIFICATION FOR GRAPHICS
 854      4A47      . . .      ; IF A GRAPHICS CURSOR KEY IS PRESSED (INDICATED
 855      4A47      . . .      ; BY CY FLAG) DONT PROCESS IT ANY FURTHER.
 856      4A47      . . .      ; THE REPEAT TIMER AND CURRENT REPEATING KEY WILL
 857      4A47      . . .      ; NOT BE UPDATED.
 858      4A47      . . .      ;*****
 859      4A47      CD 1C 4C      CALL GTKYNM      ;COMPUTE KEY NUMBER
 860      4A4A      3A 12 FF      LDA KBCTSH      ;GET CONTROL/SHIFT KEY COLUM
 861      4A4D      E6 18 .      ANI LSHFKY+RSHFKY ;SHIFT KEY DOWN
 862      4A4F      21 37 48      LXI H,LWRASC      ;(SET FOR LOWER CASE TABLE
 863      4A52      CA 58 4A      JZ GTK210      ;YES - GET THE KEY CODE
 864      4A55      21 A7 48      LXI H,UPRASC      ;NO - USE UPPER CASE TABLE
 865      4A58      . . .      ;
 866      4A58      . . .      ; EXTRACT KEY CODE FROM TABLE
 867      4A58      . . .      ;
 868      4A58      . . .      GTK210 EQU $      ;(D = 0, E = KEY NUMBER)
 869      4A58      19 . .      DAD D      ;INDEX TABLE BY KEY NUMBER
 870      4A59      46 . .      MOV R,M      ;FETCH THE TABLE VALUE
 871      4A5A      78 . .      MOV A,B
 872      4A5B      . . .      ;*****
 873      4A5B      CD 0F 60      CALL ZGCKYS      ;CHECK FOR G-CURSOR KEY
 874      4A5E      DA C7 49      JC GTK060      ;IT IS--STOP PROCESSING
 875      4A61      21 EC 91      LXI H,KBTIMR      ;ITS NOT--NOW CLEAR THE
 876      4A64      72 . .      MOV M,D      ;REPEAT TIMER (D=0)
 877      4A65      21 0D FF      LXI H,KBKNSV      ;UPDATE THE CURRENT REPEATIN
 878      4A68      73 . .      MOV M,E      ;KEY
 879      4A69      . . .      ;*****
 880      4A69      B7 . .      ORA A      ;ANY KEY CODE?
 881      4A6A      CA 8E 49      JZ GTK010      ;NO - TRY FOR ANOTHER KEY
 882      4A6D      FA C1 4A      JM GTK300      ;PROCESS FUNCTION CODE IF S=
 883      4A70      3A 12 FF      LDA KBCTSH      ;(GET CONTROL/SHIFT COLUMN
 884      4A73      E6 01 .      ANI CTLKEY      ;CONTROL KEY PRESSED?
 885      4A75      CA 92 4A      JZ GTK220      ;NO - CHECK CAPS LOCK
 886      4A78      78 . .      MOV A,B      ;YES - MOVE CHARACTER TO A
 887      4A79      FE 08 .      CPI BKSPCE      ;IS IT THE BACKSPACE KEY?
 888      4A7B      CA 8D 4A      JZ GTK215      ;YES - CHANGE TO BACK TAB
 889      4A7E      FE 09 .      CPI TAB      ;IS IT THE TAB KEY?
 890      4A80      CA 8D 4A      JZ GTK215      ;YES - CHANGE TO BACK TAB
 891      4A83      FE 40 .      CPI UPRLIM      ;IS CHARACTER ALPHABETIC?
 892      4A85      FA A7 4A      JM GTK230      ;NO - RETURN UNALTERED CHAR
 893      4A88      E6 1F .      ANI CTLSK      ;YES - MASK FOR CONTROL CODE
 894      4A8A      C3 A7 4A      JMP GTK230      ;RETURN KEY HIT
 895      4A8D      . . .      ;
 896      4A8D      . . .      ; CONTROL-BACKSPACE/TAB - CHANGE TO BACK TAB
 897      4A8D      . . .      ;
=====

```

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 27
=====
 898     4A8D      .   .   .   GTK215 EQU $
 899     4A8D     3E  E9   .   MVI  A,BACKTB ;RETURN BACK TAB CODE
 900     4A8F     C3  A7  4A   JMP  GTK230
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 28
=====
902      4A92      .      .      .      ;
903      4A92      .      .      .      ; CONTROL KEY UP - CHECK FOR CAPS LOCK
904      4A92      .      .      .      ;
905      4A92      .      .      .      GTK220 EQU $
906      4A92      3A      F3      FF      LDA      MDFLG2      ;GET TERMINAL MODE FLAGS 2
907      4A95      E6      01      .      ANI      CAPSLK      ;CAPS LOCK SET?
908      4A97      78      .      .      MOV      A,B          ;(PUT KEY CODE IN A-REG)
909      4A98      CA      A7      4A      JZ       GTK230      ;NO - RETURN KEY HIT
910      4A98      FE      60      .      CPI      LWRLIM      ;IS KEY LOWER CASE?
911      4A9D      FA      A7      4A      JM       GTK230      ;NO - RETURN KEY HIT
912      4AA0      FE      7F      .      CPI      DEL         ;IS KEY = DELETE (RUBOUT)?
913      4AA2      CA      A7      4A      JZ       GTK230      ;YES - RETURN KEY HIT
914      4AA5      D6      20      .      SUI      CPSADJ      ;NO - ADJUST TO UPPER CASE
915      4AA7      .      .      .      ;
916      4AA7      .      .      .      ; RETURN WITH LONG REPEAT DELAY
917      4AA7      .      .      .      ;
918      4AA7      .      .      .      GTK230 EQU $
919      4AA7      21      EC      91      LXI      H,KBTIMR    ;SET KEYBAORD REPEAT TIMER
920      4AAA      36      5B      .      MVI      M,LNGDLY    ;FOR LONG START DELAY
921      4AAC      .      .      .      ;
922      4AAC      .      .      .      ; GTKYX1 - RETURN KEY HIT
923      4AAC      .      .      .      ;
924      4AAC      .      .      .      GTKYX1 EQU $
925      4AAC      32      0F      FF      STA      KBCHAR      ;SAVE THE CHARACTER FOR RPT
926      4AAF      6F      .      .      MOV      L,A          ;SAVE CHARACTER IN L-REGISTE
927      4AB0      3A      11      FF      LDA      KBFLGS      ;GET KEYBOARD FLAGS
928      4AB3      E6      01      .      ANI      KBLOCK      ;KEYBOARD LOCKED?
929      4AB5      7D      .      .      MOV      A,L          ;(RECALL KEYBOARD CHAR)
930      4AB6      C2      BB      4A      JNZ      GTK240      ;YES - RETURN NO KEY HIT
931      4AB9      E1      .      .      POP      H           ;NO - RESTORE H,L
932      4ABA      C9      .      .      RET                          ;RETURN Z TRUE
933      4AB8      .      .      .      ;
934      4AB8      .      .      .      ; RETURN NO KEY HIT FOR LOCKED KEYBOARD
935      4ABB      .      .      .      ;
936      4ABB      .      .      .      GTK240 EQU $
937      4ABB      21      EC      91      LXI      H,KBTIMR    ;CLEAR THE REPEAT TIMER
938      4ABE      72      .      .      MOV      M,D
939      4ABF      E1      .      .      POP      H           ;RECALL H,L
940      4AC0      C9      .      .      RET                          ;RETURN Z FALSE
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 29
942	4AC1	.	.	*****	
943	4AC1	.	.	; FUNCTION KEY - CHECK FOR CONTROL EXCHANGES *	
944	4AC1	.	.	*****	
945	4AC1	.	.	GTK300 EQU \$	
946	4AC1	FE	87	CPI FNCLIM ;INTERNAL FUNCTION CODE?	
947	4AC3	FA	21 4B	JM GTK400 ;YES - PROCESS INTERNAL CODE	
948	4AC6	3A	12 FF	LDA KBCTSH ;NO - GET CONTROL/SHIFT COLM	
949	4AC9	E6	01	ANI CTLKEY ;CONTROL KEY DOWN?	
950	4ACB	C2	E1 4A	JNZ GTK310 ;YES - CHECK FOR ALTERNATES	
951	4ACE	3A	FA FF	LDA KBJMP2 ;NO - GET KEYBOARD JUMPERS 2	
952	4AD1	E6	08	ANI EDTWRP ;EDIT WRAP AROUND REVERSED?	
953	4AD3	78	.	MOV A,B ;(RECALL FUNCTION CODE)	
954	4AD4	CA	FF 4A	JZ GTK350 ;NO - RETURN NORMAL CODE	
955	4AD7	FE	D0	CPI DELCHR ;IS IT DELETE CHARACTER?	
956	4AD9	C2	FF 4A	JNZ GTK350 ;NO - RETURN NORMAL CODE	
957	4ADC	3E	CF	MVI A,DCHWRP ;YES - USE DELETE WRAPAROUND	
958	4ADE	C3	FF 4A	JMP GTK350 ;SET THE FUNCTION CODE	


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 31
=====
  994     4805     . . .      ;
  995     4805     . . .      ; TOGGLE SOFT KEY/NORMAL DISPLAY
  996     4805     . . .      ;
  997     4805     . . .      GTK370 EQU $
  998     4805     3A F8 FF      LDA CMFLGS      ;GET COMMON FLAGS
  999     4808     E6 08 .      ANI DEFSKY      ;SOFT KEY DEFINE ACTIVE?
1000     480A     3E EA .      MVI A,SETSFK    ;(SET CODE FOR SOFT KEYS)
1001     480C     CA FF 4A     JZ  GTK350      ;NO - RETURN NORMAL CODE
1002     480F     3C . .      INR  A          ;YES - RETURN CODE TO RESTOR
1003     4810     C3 FF 4A     JMP  GTK350      ;NORMAL DISPLAY
1004     4813     . . .      ;
1005     4813     . . .      ; DELETE CHARACTER AND CONTROL KEY DOWN
1006     4813     . . .      ;
1007     4813     . . .      GTK380 EQU $
1008     4813     3A FA FF      LDA KBJMP2      ;GET KEYBOARD JUMPERS 2
1009     4816     E6 08 .      ANI EDTWRP      ;EDIT WRAP AROUND REVERSED?
1010     4818     78 . .      MOV  A,B        ;(RECALL NORMAL CODE)
1011     4819     C2 FF 4A     JNZ  GTK350      ;YES - RETURN NORMAL CODE
1012     481C     3E CF .      MVI A,DCHWRP   ;NO - RETURN WRAP AROUND COD
1013     481E     C3 FF 4A     JMP  GTK350
=====
  
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 32
1015	4B21	. . .	;*****	
1016	4B21	. . .	; PROCESS INTERNAL FUNCTIONS *	
1017	4B21	. . .	;*****	
1018	4B21	. . .	GTK400 EQU \$	
1019	4B21	D6 80 .	SUI FNBASE ;COMPUTE FUNCTION INDEX	
1020	4B23	87 . .	ADD A	
1021	4B24	5F . .	MOV E,A ;PUT INTO E (D = 0)	
1022	4B25	21 35 49	LXI H,FNCTAB ;GET POINTER TO FUNCTION	
1023	4B28	19 . .	DAD D ;ROUTINE	
1024	4B29	7E . .	MOV A,M	
1025	4B2A	23 . .	INX H	
1026	4B2B	66 . .	MOV H,M	
1027	4B2C	6F . .	MOV L,A	
1028	4B2D	E9 . .	PCHL ;PERFORM FUNCTION	
1029	4B2E	. . .	;	
1030	4B2E	. . .	; MEMORY LOCK	
1031	4B2E	. . .	;	
1032	4B2E	. . .	CKMLOK EQU \$	
1033	4B2E	3A F4 FF	LDA MDFLG1 ;GET TERMINAL MODE FLAGS	
1034	4B31	E6 04 .	ANI MEMLOK ;MEMORY LOCK ON?	
1035	4B33	3E EC .	MVI A,MLKON ;(SET TURN ON ESCAPE CODE)	
1036	4B35	CA AC 4A	JZ GTKYX1 ;NO - RETURN TURN ON CODE	
1037	4B38	3C . .	INR A ;YES - RETURN TURN OFF CODE	
1038	4B39	C3 AC 4A	JMP GTKYX1	
1039	4B3C	. . .	;	
1040	4B3C	. . .	; DISPLAY FUNCTIONS	
1041	4B3C	. . .	;	
1042	4B3C	. . .	CKDSFN EQU \$	
1043	4B3C	3A F4 FF	LDA MDFLG1 ;GET TERMINAL MODE FLAGS 1	
1044	4B3F	E6 01 .	ANI DSPFNC ;DISPLAY FUNCTIONS ON?	
1045	4B41	C2 53 4B	JNZ GTK410 ;YES - TURN OFF DISPLAY FUNC	
1046	4B44	3A 12 FF	LDA KBCTSH ;NO - GET CONTROL/SHIFT COLM	
1047	4B47	E6 01 .	ANI CTLKEY ;CONTROL KEY DOWN?	
1048	4B49	3E D9 .	MVI A,DSPFON ;(SET CODE FOR DSP FNCT ON	
1049	4B4B	CA AC 4A	JZ GTKYX1 ;NO - TURN DISPLAY FUNCTIONS	
1050	4B4E	3E F9 .	MVI A,MNMDON ;YES - SET TO TURN ON MONITU	
1051	4B50	C3 AC 4A	JMP GTKYX1 ;MODE	
1052	4B53	. . .	;	
1053	4B53	. . .	GTK410 EQU \$	
1054	4B53	3E 9A .	MVI A,DFNCOF ;RETURN DISPLAY FUNCTIONS	
1055	4B55	C3 AC 4A	JMP GTKYX1 ;OFF CODE	

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS
1057	4B58	.	.	.	;
1058	4B58	.	.	.	; INSERT CHARACTER
1059	4B58	.	.	.	;
1060	4B58	.	.	.	CKICHR EQU \$
1061	4B58	3A	F4	FF	LDA MDFLG1 ;GET TERMINAL MODE FLAGS
1062	4B58	E6	02	.	ANI INSCR ;INSERT CHARACTER ON?
1063	4B5D	3E	D2	.	MVI A,ICHROF ;(SET OFF ESCAPE CODE)
1064	4B5F	C2	AC	4A	JNZ GTKYX1 ;YES - RETURN OFF CODE
1065	4B62	3A	FA	FF	LDA KBJMP2 ;NO - GET KEYBOARD JUMPERS 2
1066	4B65	E6	08	.	ANI EDTWRP ;REVERSE SENSE OF EDIT WRAP?
1067	4B67	CA	6C	4B	JZ CKC010 ;NO - USE NORMAL SENSE
1068	4B6A	3E	FF	.	MVI A,377Q ;YES - INVERT SENSE OF CTL
1069	4B6C	.	.	.	CKC010 EQU \$
1070	4B6C	21	12	FF	LXI H,KBCTSH ;GET CONTROL/SHIFT COLUMN
1071	4B6F	AE	.	.	XRA M ;SET FOR PROPER SENSE
1072	4B70	E6	01	.	ANI CTLKEY ;USE ALTERNATE CODE?
1073	4B72	3E	D1	.	MVI A,ICHRON ;(SET TURN ON ESCAPE CODE)
1074	4B74	CA	AC	4A	JZ GTKYX1 ;NO - RETURN NORMAL ON CODE
1075	4B77	3E	CE	.	MVI A,IWRPON ;YES - RETURN WRAP AROUND ON
1076	4B79	C3	AC	4A	JMP GTKYX1
1077	4B7C	.	.	.	;
1078	4B7C	.	.	.	; SET CAPS LOCK
1079	4B7C	.	.	.	;
1080	4B7C	.	.	.	STCPLK EQU \$
1081	4B7C	3E	01	.	MVI A,CAPSLK ;SET FLAG BIT TO BE SET
1082	4B7E	C3	8D	4B	JMP GTK450 ;SET THE FLAG AND EXIT
1083	4B81	.	.	.	;
1084	4B81	.	.	.	; SET AUTO LINE FEED
1085	4B81	.	.	.	;
1086	4B81	.	.	.	STAULF EQU \$
1087	4B81	3E	04	.	MVI A,AUTOLF ;SET FLAG BIT TO BE SET
1088	4B83	C3	8D	4B	JMP GTK450 ;SET THE FLAG AND EXIT
1089	4B86	.	.	.	;
1090	4B86	.	.	.	; SET BLOCK MODE
1091	4B86	.	.	.	;
1092	4B86	.	.	.	STBLKM EQU \$
1093	4B86	3E	02	.	MVI A,BLKMODE ;SET FLAG BIT TO BE SET
1094	4B88	C3	8D	4B	JMP GTK450 ;SET FLAG AND EXIT
1095	4B8B	.	.	.	;
1096	4B8B	.	.	.	; SET REMOTE MODE
1097	4B8B	.	.	.	;
1098	4B8B	.	.	.	STRMMD EQU \$
1099	4B8B	3E	08	.	MVI A,REMOTE ;SET FLAG BIT TO BE SET
1100	4B8D	.	.	.	;
1101	4B8D	.	.	.	GTK450 EQU \$
1102	4B8D	21	F3	FF	LXI H,MDFLG2
1103	4B90	B6	.	.	ORA M ;SET THE FLAG
1104	4B91	77	.	.	MOV M,A ;UPDATE THE FLAGS
1105	4B92	AF	.	.	XRA A ;CLEAR KEY NUMBER SAVE WORD
1106	4B93	32	0D	FF	STA KBKNSV

13255

2648A MICROCODE LISTING 'KG14'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
1107     4B96     C3  8E  49          JMP  GTK010      ;TRY FOR ANOTHER KEY
=====
```

PAGE 34

```

=====
ITEM      LOC  OBJECT CODE  SOURCE STATEMENTS                                     PAGE 35
=====
1109     4B99  . . . ;
1110     4B99  . . . ; * * * * *
1111     4B99  . . . ;
1112     4B99  . . . ; KBMON - MONITOR KEYBOARD
1113     4B99  . . . ;
1114     4B99  . . . ; ENTRY DON'T CARE
1115     4B99  . . . ;
1116     4B99  . . . ; EXIT ALL REGISTERS DESTROYED
1117     4B99  . . . ;
1118     4B99  . . . KBMON EQU $
1119     4B99  . . . ;*****
1120     4B99  . . . ; MODIFICATION FOR GRAPHICS
1121     4B99  . . . ; PERFORM TIMER INTERRUPT FUNCTIONS FOR GRAPHICS
1122     4B99  CD 14 60 CALL ZTINT ;GRAPHICS TIMER INTERRUPT
1123     4B9C  . . . ;*****
1124     4B9C  21 EC 91 LXI H,KBTIMR ;DECREMENT THE REPEAT
1125     4B9F  7E . . MOV A,M ;KEY TIMER
1126     4BA0  3D . . DCR A ;TIMER ACTIVE?
1127     4BA1  FA A8 4B JM KBM010 ;NO - DON'T UPDATE TIMER
1128     4BA4  CA A8 4B JZ KBM010 ;TIME OUT - DON'T UPDATE
1129     4BA7  77 . . MOV M,A ;YES - UPDATE TIMER
1130     4BA8  . . . KBM010 EQU $
1131     4BA8  23 . . INX H ;DECREMENT BLINK TIMER
1132     4BA9  35 . . DCR M ;TIME TO BLINK LED'S?
1133     4BAA  3A 0C FF LDA KBLEDS ;(GET CURRENT LED STATE)
1134     4BAD  F2 BA 4B JP KBM020 ;NO - SET LED'S ONLY
1135     4BB0  47 . . MOV B,A ;SAVE CURRENT STATE IN B-REG
1136     4BB1  36 1E . MVI M,BLKDLY ;YES - RESET TIMER
1137     4BB3  3A 0E FF LDA BLKFLG ;GET LED BLINK FLAG
1138     4BB6  A8 . . XRA B ;SET NEW LED VALUES
1139     4BB7  32 0C FF STA KBLEDS ;STORE NEW VALUES
1140     4BBA  . . . KBM020 EQU $
1141     4BBA  FB . . EI ;RE-ENABLE INTERRUPTS
1142     4BBB  32 00 83 STA IOKBLD ;SET KEYBOARD LED'S
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 36
=====
1144     4BBE      . . .      ;
1145     4BBE      . . .      ; CHECK FOR NEW KEY HIT
1146     4BBE      . . .      ;
1147     4BBE      . . .      KBMON1 EQU $
1148     4BBE      F3 . .      DI ;DISABLE INTERRUPTS
1149     4BBF      2A EE 91    LHL D KEYCOL ;GET CURRENT KEY COLUMNN
1150     4BC2      EB . .      XCHG ;IN D,E
1151     4BC3      2A F0 91    LHL D KBBFPT ;GET STATE TABLE POINTER
1152     4BC6      7E . .      MOV A,M ;GET PREVIOUS STATE
1153     4BC7      47 . .      MOV B,A ;SAVE IT IN B-REGISTER
1154     4BC8      32 20 83    STA IOKBCL ;SET HYSTERESIS FOR DETECTOR
1155     4BCB      1A . .      LDAX D ;GET NEW STATE
1156     4BCC      A8 . .      XRA B ;ANY CHANGES?
1157     4BCD      C2 F5 4B    JNZ KBM100 ;YES - PROCESS NEW STATE
1158     4BD0      2B . .      DCX H ;NO - DECREMENT TO NEXT COL
1159     4BD1      1D . .      DCR E ;ALL COLUMNS DONE?
1160     4BD2      F2 09 4B    JP KBM030 ;NO - CHECK NEXT COLUMN
1161     4BD5      1E 0D .      MVI E,NUMCOL ;YES - RESET COLUMN POINTERS
1162     4BD7      2E FF .      MVI L,KBBUFL+NUMCOL
1163     4BD9      . . .      KBM030 EQU $
1164     4BD9      7E . .      MOV A,M ;GET PREVIOUS STATE
1165     4BDA      47 . .      MOV B,A
1166     4BD8      32 20 83    STA IOKBCL ;SET HYSTERESIS
1167     4BDE      1A . .      LDAX D ;GET NEW STATE
1168     4BDF      A8 . .      XRA B ;ANY CHANGES?
1169     4BE0      C2 F5 4B    JNZ KBM100 ;YES - PROCESS NEW STATE
1170     4BE3      2B . .      DCX H ;NO - DECREMENT TO NEXT COL
1171     4BE4      1D . .      DCR E ;ALL COLUMNS DONE?
1172     4BE5      F2 EC 4B    JP KBM040 ;NO - EXIT
1173     4BE8      1E 0D .      MVI E,NUMCOL ;YES - RESET COLUMN POINTERS
1174     4BEA      2E FF .      MVI L,KBBUFL+NUMCOL
1175     4BEC      . . .      KBM040 EQU $
1176     4BEC      22 F0 91    SHLD KBBFPT ;SAVE COLUMN POINTERS
1177     4BEF      EB . .      XCHG
1178     4BF0      22 EE 91    SHLD KEYCOL
1179     4BF3      FB . .      EI ;RE-ENABLE INTERRUPTS
1180     4BF4      C9 . .      RET ;RETURN
=====

```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 37
1182	4BF5	.	.	;	
1183	4BF5	.	.	; KEYBOARD STATE CHANGE - ADD CHANGE TO BUFFER	
1184	4BF5	.	.	;	
1185	4BF5	.	.	KBM100 EQU \$	
1186	4BF5	22	F0 91	SHLD KBBFPT ;SAVE COLUMN POINTERS	
1187	4BF8	EB	.	XCHG	
1188	4BF9	22	EE 91	SHLD KEYCOL	
1189	4BFC	4D	.	MOV C,L ;PUT COLUMN NUMBER IN C	
1190	4BFD	2A	C2 91	LHLD KBPTPT ;GET PUT POINTER	
1191	4C00	71	.	MOV M,C ;STORE COLUMN NUMBER	
1192	4C01	2B	.	DCX H ;DECREMENT TO NEXT POSITION	
1193	4C02	A8	.	XRA B ;RESTORE NEW STATE	
1194	4C03	77	.	MOV M,A ;STORE NEW STATE IN BUFFER	
1195	4C04	47	.	MOV B,A ;SAVE CHANGED BITS IN B-REG	
1196	4C05	3E	C4 .	MVI A,KEYBFL	
1197	4C07	BD	.	CMP L ;REACHED END OF BUFFER?	
1198	4C08	C2	0D 4C	JNZ KBM110 ;NO - CHECK FOR BUFFER FULL	
1199	4C08	2E	EC .	MVI L,KEYBFL+KEYBLN ;YES - RESET POINTER	
1200	4C0D	.	.	KBM110 EQU \$	
1201	4C0D	2B	.	DCX H ;DECREMENT TO NEXT POSITION	
1202	4C0E	3A	C0 91	LDA KBGTPT	
1203	4C11	BD	.	CMP L ;BUFFER FULL?	
1204	4C12	CA	1A 4C	JZ KBM120 ;YES - DON'T UPDATE POINTERS	
1205	4C15	22	C2 91	SHLD KBPTPT ;NO - UPDATE POINTERS	
1206	4C18	EB	.	XCHG	
1207	4C19	70	.	MOV M,B ;UPDATE STATE TABLE	
1208	4C1A	.	.	KBM120 EQU \$	
1209	4C1A	FB	.	EI ;RE-ENABLE INTERRUPTS	
1210	4C1B	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 38
=====
1212     4C1C      . . .      ;
1213     4C1C      . . .      ; * * * * *
1214     4C1C      . . .      ;
1215     4C1C      . . .      ;   GTKYNM - GET KEY NUMBER
1216     4C1C      . . .      ;
1217     4C1C      . . .      ;   ENTRY   B = KEY BIT IN COLUMN
1218     4C1C      . . .      ;           E = COLUMN NUMBER
1219     4C1C      . . .      ;
1220     4C1C      . . .      ;   EXIT    A = E = KEY NUMBER
1221     4C1C      . . .      ;           L = KBKNSV-KBSTOR
1222     4C1C      . . .      ;
1223     4C1C      . . .      ;   GTKYNM EQU $
1224     4C1C      7B . . .      MOV   A,E       ;MULTIPLY COLUMN NUMBER
1225     4C1D      07 . . .      RLC                      ;BY 8
1226     4C1E      07 . . .      RLC
1227     4C1F      07 . . .      RLC
1228     4C20      5F . . .      MOV   E,A       ;PUT PRODUCT INTO E
1229     4C21      1D . . .      DCR   E         ;ADJUST TO INITIAL VALUE
1230     4C22      78 . . .      MOV   A,B       ;PUT KEY BIT IN A
1231     4C23      . . .      ;
1232     4C23      . . .      ;   ADD IN BIT NUMBER
1233     4C23      . . .      ;
1234     4C23      . . .      ;   GTN010 EQU $
1235     4C23      1C . . .      INR   E         ;INCREMENT COUNT
1236     4C24      0F . . .      RRC                      ;BIT FOUND?
1237     4C25      D2 23 4C      JNC   GTN010    ;NO - CONTINUE COUNTING
1238     4C28      7B . . .      MOV   A,E       ;YES - PUT RESULT IN A
1239     4C29      21 0D FF      LXI   H,KBKNSV ;SET H,L TO REPEATING KEY #
1240     4C2C      C9 . . .      RET                      ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 39
1242	4C2D	.	.	*****	
1243	4C2D	.	.	; SETRPT - SET REPEAT TIMER FOR FUNCTION KEYS *	
1244	4C2D	.	.	*****	
1245	4C2D	.	.	; ENTRY A = KEYBOARD CHARACTER	
1246	4C2D	.	.	;	
1247	4C2D	.	.	; EXIT KBTIMR = DELAY FOR REPEAT	
1248	4C2D	.	.	; B,H,L DESTROYED	
1249	4C2D	.	.	;	
1250	4C2D	.	.	SETRPT EQU \$	
1251	4C2D	21	EC 91	LXI H,KBTIMR ;SET H,L TO REPEAT TIMER	
1252	4C30	06	5B .	MVI B,LANGDLY ;SET B FOR LONG DELAY TIME	
1253	4C32	FE	98 .	CPI ENTRCD ;ENTER KEY?	
1254	4C34	CA	50 4C	JZ STR020 ;YES - SET FOR LONG DELAY	
1255	4C37	.	.	*****	
1256	4C37	.	.	; ADDED FOR GRAPHICS	
1257	4C37	FE	98 .	CPI GRAFUN+1 ;GRAPHICS FUNCTION?	
1258	4C39	DA	50 4C	JC STR020 ;YES, SET LONG DELAY	
1259	4C3C	.	.	*****	
1260	4C3C	FE	A1 .	CPI EXFNLM ;EXTERNAL FUNCTION?	
1261	4C3E	F8	. .	RM ;YES - SET NO REPEAT	
1262	4C3F	70	. .	MOV M,B ;SET FOR LONG START DELAY	
1263	4C40	FE	C1 .	CPI 301Q ;CURSOR CONTROL?	
1264	4C42	F8	. .	RM ;NO - RETURN LONG DELAY	
1265	4C43	FE	C5 .	CPI 305Q	
1266	4C45	FA	4E 4C	JM STR010 ;YES - SET SHORT DELAY	
1267	4C48	FE	D3 .	CPI 3230 ;ROLL UP OR ROLL DOWN?	
1268	4C4A	F8	. .	RM ;NO - RETURN LONG DELAY	
1269	4C4B	FE	D5 .	CPI 325Q	
1270	4C4D	F0	. .	RP ;NO - RETURN LONG DELAY	
1271	4C4E	.	.	STR010 EQU \$;YES - SET FOR SHORT REPEAT	
1272	4C4E	06	33 .	MVI B,SRTDLY ;START DELAY	
1273	4C50	.	.	STR020 EQU \$	
1274	4C50	70	. .	MOV M,B ;SET REPEAT TIMER	
1275	4C51	C9	. .	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 40
=====
1277     4C52      . . .      ;*****
1278     4C52      . . .      ; KBCTL - PERFORM KEYBOARD CONTROL FUNCTION *
1279     4C52      . . .      ;*****
1280     4C52      . . .      ;
1281     4C52      . . .      ; ENTRY   A = CONTROL CODE
1282     4C52      . . .      ;
1283     4C52      . . .      ; EXIT    DETERMINED BY INDIVIDUAL CONTROL
1284     4C52      . . .      ;          ROUTINES
1285     4C52      . . .      ;          GENERALLY D-L REGISTERS ARE SAVED
1286     4C52      . . .      ;          AND A-C DESTROYED
1287     4C52      . . .      ;
1288     4C52      . . .      KBCTL EQU $
1289     4C52      FE 0B      CPI  KBCTLM+1 ;CONTROL CODE WITHIN RANGE?
1290     4C54      D0 . .      RNC          ;NO - EXIT IMMEDIATELY
1291     4C55      E5 . .      PUSH H       ;YES - SAVE THE WORKING
1292     4C56      D5 . .      PUSH D       ;REGISTERS
1293     4C57      87 . .      ADD  A       ;DOUBLE THE PARAMETER VALUE
1294     4C58      5F . .      MOV  E,A     ;COMPUTE CONTROL VECTOR
1295     4C59      16 00      MVI  D,0     ;LOCATION
1296     4C5B      21 63 4C   LXI  H,KBCTAB-2
1297     4C5E      19 . .      DAD  D
1298     4C5F      5E . .      MOV  E,M     ;FETCH THE CONTROL VECTOR
1299     4C60      23 . .      INX  H
1300     4C61      66 . .      MOV  H,M
1301     4C62      6B . .      MOV  L,E
1302     4C63      D1 . .      POP  D       ;RECALL D AND E
1303     4C64      E9 . .      PCHL        ;GO TO CONTROL ROUTINE
1304     4C65      . . .      ;
1305     4C65      . . .      ; CONTROL VECTORS
1306     4C65      . . .      ;
1307     4C65      . . .      KBCTAB EQU $
1308     4C65      79 4C      DW  LOKKBD   ;1 - LOCK KEYBOARD
1309     4C67      89 4C      DW  UNLKBD   ;2 - UNLOCK KEYBOARD
1310     4C69      AB 4C      DW  RPTKEY   ;3 - REPEAT LAST KEY HIT
1311     4C6B      92 4C      DW  STBLMD   ;4 - SET PERMANENT BLOCK MOD
1312     4C6D      B4 4C      DW  STRTST   ;5 - SELF-TEST START
1313     4C6F      DB 4C      DW  ENDTST   ;6 - SELF-TEST END
1314     4C71      EA 4C      DW  RSETKB   ;7 - RESET KEYBOARD
1315     4C73      2B 4D      DW  CKIOKY   ;8 - CHECK FOR I/O KEY
1316     4C75      5E 4D      DW  STPRPT   ;9 - STOP KEY REPEAT
1317     4C77      64 4D      DW  CKBRKY   ;10 - CHECK FOR BREAK KEY
1318     4C79      . . .      ;*****
1319     4C79      . . .      ; UNSUPPORTED CONTROL FUNCTIONS *
1320     4C79      . . .      ;*****
1321     4C79      . . .      ; DFAD SWCHAR 11 - SWITCH CHARACTER SET
1322     4C79      . . .      ; DFAD SETFRN 12 - UPDATE FOREIGN MODE
1323     4C79      . . .      ; DFAD STCHST 13 - SET FOREIGN MODE OUTPUT
1324     4C79      . . .      ; DFAD FRNMD1 14 - SET FOREIGN MODE 1
1325     4C79      . . .      ; DFAD FRNMD2 15 - SET FOREIGN MODE 2
1326     4C79      . . .      ;
=====

```

13255
2648A MICROCODE LISTING 'KG14'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC   OBJECT CODE  SOURCE STATEMENTS                                PAGE 41
=====
1327     0005   . . .      ZSTRTS EQU 5      ;SELF-TEST START CONTROL
1328     000A   . . .      KBCTLM EQU 10     ;KEYBOARD CONTROL LIMIT
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
1330      4C79      . . .      ;
1331      4C79      . . .      ; LOKKBD - LOCK THE KEYBOARD
1332      4C79      . . .      ;
1333      4C79      . . .      LOKKBD EQU $
1334      4C79      AF . . .      XRA A          ;CLEAR THE REPEAT TIMER
1335      4C7A      32 EC 91      STA KBTIMR
1336      4C7D      3A 11 FF      LDA KBFLGS
1337      4C80      E6 F7 .      ANI 377Q-RPTKY ;CLEAR REPEAT KEY FLAG
1338      4C82      F6 01 .      ORI KBLOCK     ;AND SET LOCKED FLAG
1339      4C84      32 11 FF      STA KBFLGS
1340      4C87      . . .      ;
1341      4C87      . . .      KBCTX1 EQU $
1342      4C87      E1 . . .      POP H          ;RESTORE H AND L
1343      4C88      C9 . . .      RET           ;RETURN
1344      4C89      . . .      ;
1345      4C89      . . .      ; UNLKBD - UNLOCK KEYBOARD
1346      4C89      . . .      ;
1347      4C89      . . .      UNLKBD EQU $
1348      4C89      3E FE .      MVI A,377Q-KBLOCK ;CLEAR KEYBOARD
1349      4C8B      21 11 FF      LXI H,KBFLGS  ;CLEAR KEYBOARD FLAG
1350      4C8E      A6 . . .      ANA M
1351      4C8F      77 . . .      MOV M,A
1352      4C90      E1 . . .      POP H          ;RESTORE H AND L
1353      4C91      C9 . . .      RET           ;RETURN
1354      4C92      . . .      ;
1355      4C92      . . .      ; STBLMD - SET PERMANENT BLOCK MODE
1356      4C92      . . .      ;
1357      4C92      . . .      STBLMD EQU $
1358      4C92      21 11 FF      LXI H,KBFLGS  ;SET PERMANENT BLOCK MODE
1359      4C95      7E . . .      MOV A,M       ;FLAG
1360      4C96      F6 02 .      ORI PERMBM
1361      4C98      77 . . .      MOV M,A
1362      4C99      21 F3 FF      LXI H,MDFLG2  ;SET BLOCK MODE FLAG
1363      4C9C      7E . . .      MOV A,M
1364      4C9D      F6 02 .      ORI BLKMDE
1365      4C9F      77 . . .      MOV M,A
1366      4CA0      3E C0 .      MVI A,HNDSHK+DC2SND
1367      4CA2      32 10 FF      STA KBJ1MS    ;SET JUMPER INHIBIT MASK
1368      4CA5      2E FB .      MVI L,KBJMPR-CMSTOR
1369      4CA7      B6 . . .      ORA M         ;SET JUMPERS TO INHIBIT
1370      4CA8      77 . . .      MOV M,A       ;ALL TOP LEVEL HANDSHAKE
1371      4CA9      E1 . . .      POP H         ;RESTORE H AND L
1372      4CAA      C9 . . .      RET           ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1374	4CAB	.	.	;	43
1375	4CAB	.	.	; RPTKEY - REPEAT LAST KEY HIT	
1376	4CAB	.	.	;	
1377	4CAB	.	.	RPTKEY EQU \$	
1378	4CAB	3E	08	MVI A,RPTKY ;SET REPEAT KEY FLAG	
1379	4CAD	21	11	LXI H,KBFLGS ;SET KEYBOARD FLAG	
1380	4CB0	B6	.	ORA M	
1381	4CB1	77	.	MOV M,A	
1382	4CB2	E1	.	POP H ;RESTORE H AND L	
1383	4CB3	C9	.	RET ;RETURN	
1384	4CB4	.	.	*****	
1385	4CB4	.	.	; STRTST - SELF-TEST START *	
1386	4CB4	.	.	*****	
1387	4CB4	.	.	STRTST EQU \$	
1388	4CB4	21	0C	LXI H,KBLEDS ;GET CURRENT LED'S STATE	
1389	4CB7	7E	.	MOV A,M	
1390	4CB8	FE	7F	CPI 377Q-BELLED ;ALL LIGHTS ON ALREADY?	
1391	4CBA	CA	87	JZ KBCTX1 ;YES - DON'T CHANGE LED'S	
1392	4CBD	32	0B	STA LEDSAV ;NO - SAVE CURRENT LED STAT	
1393	4CC0	3E	7F	MVI A,377Q-BELLED	
1394	4CC2	77	.	MOV M,A ;FORCE ALL LED'S ON	
1395	4CC3	32	00	STA IOKBLD	
1396	4CC6	2E	F8	MVI L,CMFLGS-CMSTOR	
1397	4CC8	7E	.	MOV A,M ;TURN FORCE FULL RESET	
1398	4CC9	F6	04	ORI FRCRST ;FLAG	
1399	4CCB	77	.	MOV M,A	
1400	4CCC	E1	.	POP H ;RESTORE H AND L	
1401	4CCD	.	.	STRTS1 EQU \$;ENABLE RESET KEY	
1402	4CCD	3E	02	MVI A,RSTON	
1403	4CCF	32	80	STA IOKBCU	
1404	4CD2	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  44
=====
1406     4CD3      . . .      ;*****
1407     4CD3      . . .      ; ENDTST - END SELF-TEST *
1408     4CD3      . . .      ;*****
1409     4CD3      . . .      ENDTSO EQU $ ;CHECK FOR SOFT RESET IN
1410     4CD3      21 UC FF    LXI H,KBLEDS ;PROGRESS
1411     4CD6      7E . .     MOV A,M ;GET CURRENT LED SETTINGS
1412     4CD7      FE 7F .    CPI 377Q-BELLED ;SUFT RESET IN PROGRESS?
1413     4CD9      C0 . .     RNZ ;NO - RETURN
1414     4CDA      E5 . .     PUSH H ;YES - RESET LED'S
1415     4CDB      . . .      ENDTST EQU $
1416     4CDB      21 F8 FF    LXI H,CMFLGS
1417     4CDE      7E . .     MOV A,M ;TURN OFF FORCE FULL RESET
1418     4CDF      E6 FB .    ANI 377Q-FRCRST ;FLAG
1419     4CE1      77 . .     MOV M,A
1420     4CE2      3A 0B FF    LDA LEDSAV ;RESTORE LED STATE
1421     4CE5      32 0C FF    STA KBLEDS ;*****
1422     4CE8      . . .      ; * "KBMON" WILL RESTORE *
1423     4CE8      . . .      ; * KEYBOARD LED'S WHEN *
1424     4CE8      . . .      ; * TIMER INTERRUTS OCCURS *
1425     4CE8      . . .      ; *****
1426     4CE8      E1 . .     POP H ;RESTORE H AND L
1427     4CE9      C9 . .     RET ;RETURN
=====
  
```


ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1429	4CEA	.	.	*****	45
1430	4CEA	.	.	; RSETKB - RESET KEYBOARD FOR SOFT RESET *	
1431	4CEA	.	.	*****	
1432	4CEA	.	.	RSETKB EQU \$	
1433	4CEA	CD	5B 49	CALL INITKO ;INITIALIZE BUFFER POINTERS	
1434	4CED	21	F2 91	LXI H,KBBUF ;TRANSFER VALUES FROM "REAL	
1435	4CF0	11	12 FF	LXI D,KBBUF2 ;"TIME" STATE TABLE TO	
1436	4CF3	0E	0E .	MVI C,NUMCOL+1 ;CURRENTLY ACTIVE TABLE	
1437	4CF5	.	.	;	
1438	4CF5	.	.	RSK010 EQU \$	
1439	4CF5	7E	.	MOV A,M	
1440	4CF6	12	.	STAX D	
1441	4CF7	23	.	INX H	
1442	4CF8	13	.	INX D	
1443	4CF9	0D	.	DCR C ;ALL ENTRIES DONE?	
1444	4CFA	C2	F5 4C	JNZ RSK010 ;NO - CONTINUE TRANSFER	
1445	4CFD	.	.	;	
1446	4CFD	.	.	; CLEAR KEYBOARD FLAGS	
1447	4CFD	.	.	;	
1448	4CFD	21	11 FF	LXI H,KBFLGS ;YES - UNLOCK THE KEYBOARD	
1449	4D00	7E	.	MOV A,M	
1450	4D01	E6	F6 .	ANI 377Q-KBLOCK-RPTKY	
1451	4D03	77	.	MOV M,A	
1452	4D04	2E	0C .	MVI L,KBLEDS-KBSSTR	
1453	4D06	7E	.	MOV A,M ;TURN OFF RECORD, SELECT AND	
1454	4D07	E6	9E .	ANI 377Q-RECLEL-SELLED-DSFLED	
1455	4D09	77	.	MOV M,A ;FUNCTIONS LED'S	
1456	4D0A	2E	0E .	MVI L,BLKFLG-KBSSTR	
1457	4D0C	7E	.	MOV A,M ;STOP RECORD, SELECT AND	
1458	4D0D	E6	9E .	ANI 377Q-RECLEL-SELLED-DSFLED	
1459	4D0F	77	.	MOV M,A ;FUNCTIONS BLINKING	
1460	4D10	21	F4 FF	LXI H,MDFLG1	
1461	4D13	7E	.	MOV A,M ;TURN OFF RECORD, SELECT AND	
1462	4D14	E6	9E .	ANI 377Q-RECORD-SELECT-DSPFNC ;DISPLAY	
1463	4D16	77	.	MOV M,A ;FUNCTIONS MODE FLAGS	
1464	4D17	AF	.	XRA A	
1465	4D18	32	F6 FF	STA INTFLG ;CLEAR INTERRUPT FLAG	
1466	4D1B	32	EC 91	STA KBTIMR ;STOP KEY REPEAT	
1467	4D1E	21	F8 FF	LXI H,CMFLGS ;CLEAR THE REMOTE SET FLAG	
1468	4D21	7E	.	MOV A,M ;TO FORCE DATA COMM RESET	
1469	4D22	E6	EF .	ANI 377Q-REMSET	
1470	4D24	77	.	MOV M,A	
1471	4D25	CD	D0 4D	CALL BELL ;SOUND THE BELL	
1472	4D28	C3	B4 4C	JMP STRTST ;TURN ON ALL LED'S AND EXIT	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 46
=====
1474      4D2B      .      .      .      ;*****
1475      4D2B      .      .      .      ; CKIOKY - CHECK FOR I/O CONTROL (GREEN) KEY *
1476      4D2B      .      .      .      ;*****
1477      4D2B      .      .      .      ;
1478      4D2B      .      .      .      ; EXIT      Z - I/O CONTROL KEY NOT PRESSED
1479      4D2B      .      .      .      ;          NZ - I/O CONTROL KEY PRESSED
1480      4D2B      .      .      .      ;          A-E DESTROYED
1481      4D2B      .      .      .      ;
1482      4D2B      .      .      .      CKIOKY EQU  $
1483      4D2B      3A      0F      FF      LDA      KBCHAR      ;GET CURRENT KEYBOARD CHAR
1484      4D2E      67      .      .      MOV      H,A          ;SAVE CHARACTER IN H-REGISTE
1485      4D2F      .      .      .      ;          (CHARACTER RESTORED IN H
1486      4D2F      .      .      .      ;          ON RETURN FROM "CKBRK1")
1487      4D2F      CD      6F      4D      CALL     CKBRK1      ;CLEAR KEYBOARD BUFFER
1488      4D32      3A      18      FF      LDA      IOKYRW      ;NO - GET I/O CNTRL KEY ROW
1489      4D35      E6      01      .      ANI      IOKYCL      ;I/O CONTROL KEY DOWN?
1490      4D37      CA      87      4C      JZ       KBCTX1      ;NO - EXIT
1491      4D3A      3E      5B      .      MVI      A,LNGDLY    ;YES - RESTORE REPEAT TIMER
1492      4D3C      32      EC      91      STA      KBTIMR
1493      4D3F      7C      .      .      MOV      A,H          ;RESTORE ORIGINAL KEYBOARD
1494      4D40      21      0F      FF      LXI      H,KBCHAR    ;CHARACTER
1495      4D43      77      .      .      MOV      M,A
1496      4D44      FE      FA      .      CPI      TEST        ;CURRENT KEY = TEST?
1497      4D46      3A      18      FF      LDA      IOKYRW      ;(SET FOR ALTERNATE SELF-
1498      4D49      11      98      02      LXI      D,ENTCOL*256+ENTRCD ;TEST CHECK)
1499      4D4C      CA      55      4D      JZ       CKI010      ;YES - CHECK FOR ENTER KEY
1500      4D4F      3A      17      FF      LDA      TSTROW      ;NO - CHECK FOR TEST KEY
1501      4D52      11      FA      01      LXI      D,TSTCOL*256+TEST
1502      4D55      .      .      .      ;
1503      4D55      .      .      .      ; CHECK FOR ALTERNATING SELF-TEST
1504      4D55      .      .      .      ;
1505      4D55      .      .      .      CKI010 EQU  $
1506      4D55      2F      .      .      CMA      ;INVERT KEYBOARD ROW SETTING
1507      4D56      A2      .      .      ANA      D          ;OTHER SELF-TEST KEY DOWN?
1508      4D57      C2      87      4C      JNZ      KBCTX1      ;NO - EXIT
1509      4D5A      73      .      .      MOV      M,E          ;YES - SET ALTERNATE TEST KE
1510      4D5B      B3      .      .      ORA      E          ;SET Z FALSE
1511      4D5C      E1      .      .      POP      H          ;RESTORE H AND L
1512      4D5D      C9      .      .      RET      ;RETURN Z FALSE
1513      4D5E      .      .      .      ;*****
1514      4D5E      .      .      .      ; STPRPT - STOP KEY REPEAT *
1515      4D5E      .      .      .      ;*****
1516      4D5E      .      .      .      STPRPT EQU  $
1517      4D5E      AF      .      .      XRA      A          ;ZERO REPEAT TIMER
1518      4D5F      32      EC      91      STA      KBTIMR
1519      4D62      E1      .      .      POP      H          ;RESTORE H AND L
1520      4D63      C9      .      .      RET      ;RETURN
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 47
=====
1522     4D64      . . .      ;*****
1523     4D64      . . .      ; CHECK FOR BREAK KEY DOWN *
1524     4D64      . . .      ;*****
1525     4D64      . . .      ;
1526     4D64      . . .      ; EXIT      Z - BREAK KEY NOT DOWN
1527     4D64      . . .      ;          NZ - BREAK KEY DOWN
1528     4D64      . . .      ;
1529     4D64      . . .      CKBRKY EQU $
1530     4D64      E1 . . .      POP H          ;POP OFF H AND L
1531     4D65      AF . . .      XRA A          ;CHECK HARDWARE LATCHES
1532     4D66      32 20 83      STA IUKBCL     ;FOR BREAK KEY DOWN
1533     4D69      3A 0D 83      LDA IUKB+BRKYRN
1534     4D6C      E6 01 .      ANI BRKCOL     ;BREAK KEY DOWN?
1535     4D6E      C8 . . .      RZ             ;NO - RETURN FAIL
1536     4D6F      . . .      CKBRK1 EQU $   ;YES - CLEAR KEYBOARD BUFFER
1537     4D6F      . . .      CKB010 EQU $
1538     4D6F      CD 75 49      CALL GTKEY     ;ANY KEY HIT?
1539     4D72      3C . . .      INR A          ;(ADJUST FOR LOCKED KEYBD)
1540     4D73      C2 6F 4D      JNZ CKB010     ;YES - CONTINUE UNTIL NONE
1541     4D76      . . .      NZEXIT EQU $   ;NON-ZERO EXIT
1542     4D76      3C . . .      INR A          ;NO - FORCE Z-FALSE
1543     4D77      C9 . . .      RET           ;RETURN NZ
=====
  
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	48
1545	4D78	. . .	;		
1546	4D78	. . .	; * * * * *		
1547	4D78	. . .	;		
1548	4D78	. . .	; SETMD1 - SET MODE 1 FLAGS		
1549	4D78	. . .	;		
1550	4D78	. . .	; ENTRY A = FLAG BIT TO BE SET		
1551	4D78	. . .	; B = 3778, BLINK ASSOCIATED LED		
1552	4D78	. . .	; = 0, DON'T BLINK LED		
1553	4D78	. . .	;		
1554	4D78	. . .	; EXIT A,C DESTROYED		
1555	4D78	. . .	; ASSOCIATED LED, IF ANY, IS SET ON		
1556	4D78	. . .	;		
1557	4D78	. . .	SETMD1 EQU \$		
1558	4D78	E5 . .	PUSH H ;SAVE H,L		
1559	4D79	4F . .	MOV C,A ;SAVE BIT TO BE SET		
1560	4D7A	21 F4 FF	LXI H,MDFLG1		
1561	4D7D	B6 . .	ORA M ;SET THE FLAG		
1562	4D7E	77 . .	MOV M,A ;STORE UPDATED FLAGS		
1563	4D7F	CD BA 4D	CALL FNDLED ;LOCATE ASSOCIATED LED		
1564	4D82	4F . .	MOV C,A ;SAVE LED BIT		
1565	4D83	F3 . .	DI ;DISABLE INTERRUPTS		
1566	4D84	86 . .	ORA M ;ADD BIT TO LED CONTROL WORD		
1567	4D85	77 . .	MOV M,A ;STORE NEW CONTROL WORD		
1568	4D86	F8 . .	EI ;RE-ENABLE INTERRUPTS		
1569	4D87	79 . .	MOV A,C ;RECALL LED BIT		
1570	4D88	21 0E FF	LXI H,BLKFLG ;SET H,L TO BLINK FLAG		
1571	4D8B	04 . .	INR B ;SET LED BLINKING?		
1572	4D8C	CA 94 4D	JZ ST1010 ;YES - SET BLINK BIT		
1573	4D8F	2F . .	CMA ;NO - SET A TO CLEAR MASK		
1574	4D90	A6 . .	ANA M ;CLEAR BLINK FLAG		
1575	4D91	C3 95 4D	JMP ST1020 ;GO UPDATE BLINK FLAG		
1576	4D94	. . .	;		
1577	4D94	. . .	ST1010 EQU \$		
1578	4D94	B6 . .	ORA M ;SET BLINK BIT		
1579	4D95	. . .	ST1020 EQU \$		
1580	4D95	77 . .	MOV M,A		
1581	4D96	F3 . .	DI ;DISABLE INTERRUPTS		
1582	4D97	CD D3 4C	CALL ENDTSU ;CHECK SOFT RESET IN PROGRES		
1583	4D9A	B6 . .	ORA M ;TURN ALL BLINKING LED'S ON		
1584	4D9B	77 . .	MOV M,A		
1585	4D9C	F8 . .	EI ;RE-ENABLE INTERRUPTS		
1586	4D9D	E1 . .	POP H ;RESTORE H,L		
1587	4D9E	C9 . .	RET ;RETURN		

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
					49
1589	4D9F	.	.	;	
1590	4D9F	.	.	; * * * * *	
1591	4D9F	.	.	;	
1592	4D9F	.	.	; CLRMD1 - CLEAR MODE 1 FLAG	
1593	4D9F	.	.	;	
1594	4D9F	.	.	; ENTRY A = FLAG BIT TO BE CLEARED	
1595	4D9F	.	.	;	
1596	4D9F	.	.	; EXIT A,C DESTROYED	
1597	4D9F	.	.	; ASSOCIATED LED, IF ANY, IS SET OFF	
1598	4D9F	.	.	;	
1599	4D9F	.	.	; INTERRUPT SYSTEM IS DISABLED AND ENABLED	
1600	4D9F	.	.	; AGAIN	
1601	4D9F	.	.	;	
1602	4D9F	.	.	; CLRMD1 EQU \$	
1603	4D9F	E5	.	PUSH H ;SAVE H,L	
1604	4DA0	4F	.	MOV C,A ;SAVE BIT TO BE CLEARED	
1605	4DA1	21 F4	FF	LXI H,MDFLG1	
1606	4DA4	2F	.	CMA ;COMPLEMENT TO GET CLEAR MAS	
1607	4DA5	A6	.	ANA M ;CLEAR THE BIT	
1608	4DA6	77	.	MOV M,A	
1609	4DA7	CD BA	4D	CALL FNDLED ;LOCATE ASSOCIATED LED	
1610	4DAA	2F	.	CMA ;COMPLEMENT TO GET CLEAR MAS	
1611	4DAB	4F	.	MOV C,A ;SAVE CLEAR PATTERN	
1612	4DAC	21 0E	FF	LXI H,BLKFLG ;CLEAR BLINK FLAG IN CASE	
1613	4DAF	A6	.	ANA M ;LED WAS BLINKING	
1614	4DB0	77	.	MOV M,A	
1615	4DB1	F3	.	DI ;DISABLE INTERRUPTS	
1616	4DB2	CD D3	4C	CALL ENDTSO ;CHECK SOFT RESET IN PROGRES	
1617	4DB5	A1	.	ANA C ;CLEAR LED BIT	
1618	4DB6	77	.	MOV M,A	
1619	4DB7	FB	.	EI ;ENABLE INTERRUPTS	
1620	4DB8	E1	.	POP H ;RESTORE H,L	
1621	4DB9	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 50
=====
1623     4DBA      . . .      ;
1624     4DBA      . . .      ; * * * * *
1625     4DBA      . . .      ;
1626     4DBA      . . .      ; FNDLED - FIND LED BIT ASSOCIATED WITH
1627     4DBA      . . .      ; MODE 1 FLAGS
1628     4DBA      . . .      ;
1629     4DBA      . . .      ; ENTRY C = MODE 1 FLAG BIT TO BE USED
1630     4DBA      . . .      ;
1631     4DBA      . . .      ; EXIT A = ASSOCIATED LED BIT
1632     4DBA      . . .      ;
1633     4DBA      . . .      ; H,L = KBLEDS
1634     4DBA      . . .      ;
1635     4DBA      . . .      FNDLED EQU $
1636     4DBA      79 . .      MOV A,C ;PUT BIT INTO A
1637     4DBB      21 C7 4D      LXI H,LEDTAB-1 ;SET INITIAL TABLE ADDRESS
1638     4DBE      . . .      ;
1639     4DBE      . . .      FLD010 EQU $
1640     4DBE      23 . .      INX H ;ADVANCE TO NEXT TABLE ENTRY
1641     4DBF      0F . .      RRC ;BIT FOUND?
1642     4DC0      02 BE 4D      JNC FLD010 ;NO - GO TO NEXT ENTRY
1643     4DC3      7E . .      MOV A,M ;YES - GET TABLE ENTRY
1644     4DC4      21 0C FF      LXI H,KBLEDS ;SET H,L TO LED CONTROL WORD
1645     4DC7      C9 . .      RET ;RETURN
1646     4DC8      . . .      ;
1647     4DC8      . . .      ; LED ASSOCIATION TABLE
1648     4DC8      . . .      ;
1649     4DC8      . . .      LEDTAB EQU $ ;BIT FUNCTION
1650     4DC8      01 . .      DB DSFLED ;0 - DISPLAY FUNCTIONS
1651     4DC9      02 . .      DB ICHLED ;1 - INSERT CHARACTER
1652     4DCA      04 . .      DB MLKLED ;2 - MEMORY LOCK
1653     4DCB      00 . .      DB 0 ;3 - FORMAT MODE
1654     4DCC      10 . .      DB EDTLED ;4 - EDIT MODE
1655     4DCD      20 . .      DB SELLED ;5 - SELECT MODE
1656     4DCE      40 . .      DB RECLEL ;6 - RECORD MODE
1657     4DCF      00 . .      DB 0 ;7 - UNUSED
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	51
1659	4DD0	. . .	;		
1660	4DD0	. . .	; * * * * *		
1661	4DD0	. . .	;		
1662	4DD0	. . .	; BELL - SOUND THE KEYBOARD BELL		
1663	4DD0	. . .	;		
1664	4DD0	. . .	; ENTRY DON'T CARE		
1665	4DD0	. . .	;		
1666	4DD0	. . .	; EXIT A DESTROYED		
1667	4DD0	. . .	; Z FALSE		
1668	4DD0	. . .	;		
1669	4DD0	. . .	BELL EQU \$		
1670	4DD0	. . .	;***** GRAPHICS MODIFICATION *****		
1671	4DD0	E5 . .	PUSH H ;SAVE HL		
1672	4DD1	CD 3E 60	CALL ZTKCLR ;CLEAR ECHO SUPRESS		
1673	4DD4	E1 . .	POP H		
1674	4DD5	. . .	;*****		
1675	4DD5	3A 0C FF	LDA KBLEDS ;GET CURRENT LED SETTINGS		
1676	4DD8	F6 80 .	ORI BELLED ;ADD CONTROL TO SOUND BELL		
1677	4DDA	32 00 83	STA IOKBLD ;SOUND THE BELL		
1678	4DDD	C9 . .	RET ;RETURN		
1679	4DDE	. . .	;*****		
1680	4DDE	. . .	; SETXMT - TURN ON TRANSMIT LED *		
1681	4DDE	. . .	;*****		
1682	4DDE	. . .	;		
1683	4DDE	. . .	; EXIT Z = TRUE		
1684	4DDE	. . .	;		
1685	4DDE	. . .	SETXMT EQU \$		
1686	4DDE	3E 08 .	MVI A,XMTLED ;SET TRANSMIT LED BIT		
1687	4DE0	21 0C FF	LXI H,KBLEDS ;SET H,L TO LED STATE WORD		
1688	4DE3	B6 . .	ORA M ;ADD TRANSMIT LED BIT		
1689	4DE4	77 . .	MOV M,A ;UPDATE LED STATE		
1690	4DE5	BF . .	CMP A ;SET Z TRUE		
1691	4DE6	C9 . .	RET ;RETURN		
1692	4DE7	. . .	;*****		
1693	4DE7	. . .	; CLRXMT - TURN OFF TRANSMIT LED *		
1694	4DE7	. . .	;*****		
1695	4DE7	. . .	CLRXMT EQU \$		
1696	4DE7	21 0C FF	LXI H,KBLEDS		
1697	4DEA	7E . .	MOV A,M ;GET CURRENT LED SETTINGS		
1698	4DEB	FE 7F .	CPI 377Q-BELLED ;SET FOR TEST MODE?		
1699	4DED	C8 . .	RZ ;YES - RETURN		
1700	4DEE	E6 F7 .	ANI 377Q-XMTLED ;NO - CLEAR TRANSMIT		
1701	4DF0	77 . .	MOV M,A ;LED		
1702	4DF1	C9 . .	RET ;RETURN		

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 52
=====
1704     4DF2     . . .      ;*****
1705     4DF2     . . .      ; STJMPR - SET KEYBOARD JUMPER ESCAPE SEQUENCE *
1706     4DF2     . . .      ;*****
1707     4DF2     . . .      STJMPR EQU $
1708     4DF2     2A FA FF    LHL D KBJMP2      ;INITIALIZE PARAMETER VALUES
1709     4DF5     22 DA FF    SHLD PARM2
1710     4DF8     3A F9 FF    LDA KBJMP3
1711     4DF8     32 D9 FF    STA PARM3
1712     4DFE     21 3E 4E    LXI H,STJTAB     ;SET ESC SEQ RANGE TABLE ADD
1713     4E01     . . .      ;*****
1714     4E01     . . .      ; SET RANGE TABLE AND CLEAR INPUT ACCUMULATOR *
1715     4E01     . . .      ;*****
1716     4E01     . . .      STJMP1 EQU $
1717     4E01     3E 0A .     MVI A,DECRDX     ;SET INPUT RADIX AS DECIMAL
1718     4E03     32 D4 FF    STA RADIX
1719     4E06     22 D2 FF    SHLD RNGTA       ;SET NEW RANGE TABLE VALUE
1720     4E09     . . .      STJMP2 EQU $
1721     4E09     21 00 00    LXI H,0          ;CLEAR INPUT ACCUMULATOR
1722     4E0C     22 DE FF    SHLD IODATA
1723     4E0F     . . .      STJMP3 EQU $     ;CONTINUE ESCAPE SEQUENCE
1724     4E0F     3E 02 .     MVI A,2          ;SET ESC SEQ FLAG TO INDICAT
1725     4E11     32 D1 FF    STA ESCFLG       ;ESC SEQ IN PROGRESS
1726     4E14     C9 . .      RET              ;RETURN TO WAIT LOOP
=====
  
```



```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
1728      4E15      . . .      ;*****
1729      4E15      . . .      ; PARAMETER RECEIVED - SET ITS VALUE *
1730      4E15      . . .      ;*****
1731      4E15      . . .      ;
1732      4E15      . . .      ; ENTRY   B = 1
1733      4E15      . . .      ;         C = INPUT CHARACTER
1734      4E15      . . .      ;         H = BASEH
1735      4E15      . . .      ;
1736      4E15      . . .      STJ100 EQU $      ;ENTRY FOR INPUTS P-Z
1737      4E15      05 . . .      DCR B           ;SET ADJUSTMENT FACTOR
1738      4E16      . . .      STJ110 EQU $      ;ENTRY FOR INPUTS J-N
1739      4E16      05 . . .      DCR B           ;SET ADJUSTMENT FACTOR
1740      4E17      . . .      STJ120 EQU $      ;ENTRY FOR INPUTS A-H
1741      4E17      79 . . .      MOV A,C         ;PUT INPUT CHAR IN A-REG
1742      4E18      E6 DF .      ANI 377Q-40Q    ;FORCE TO UPPER CASE VALUE
1743      4E1A      D6 42 .      SUI A+1         ;SET TO RANGE 0-23
1744      4E1C      80 . . .      ADD B           ;ADD IN ADJUSTMENT FOR
1745      4E1D      . . .      ;               MISSING LETTERS (I, O)
1746      4E1D      CD AB 4E      CALL STPARM     ;SET THE PARAMETER
1747      4E20      79 . . .      MOV A,C         ;RECALL INPUT CHARACTER
1748      4E21      E6 20 .      ANI 40Q         ;IS IT UPPER CASE?
1749      4E23      C2 09 4E     JNZ STJMP2      ;NO - CLEAR INPUT ACCUMULATO
1750      4E26      . . .      ;               AND CONTINUE ESCAPE SEQ
1751      4E26      . . .      ;
1752      4E26      11 DB FF     LXI D,PARM1     ;NO - SET KEYBOARD STRAPS
1753      4E29      21 FB FF     LXI H,KBJMPR    ;(D,E=SOURCE; H,L=DEST)
1754      4E2C      3A 10 FF     LDA KBJ1MS      ;GET INHIBIT MASK
1755      4E2F      CD A1 4E     CALL STBITS     ;SET JUMPER 1 VALUES
1756      4E32      . . .      ;
1757      4E32      3A 06 50     LDA DCJMS2      ;GET JUMPER 2 INHIBIT MASK
1758      4E35      CD A1 4E     CALL STBITS     ;SET JUMPER 2 VALUES
1759      4E38      . . .      ;
1760      4E38      3A 05 50     LDA DCJMSK      ;GET JUMPER 3 INHIBIT MASK
1761      4E3B      . . .      ;*****
1762      4E3B      CD A1 4E     CALL STBITS     ;SET JUMPER 3 VALUES
1763      4E3E      C3 59 60     JMP ZTKSTR      ;TEST TEK MODE STRAPS
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 54
=====
1765     4E41      . . .      ;*****
1766     4E41      . . .      ; SET JUMPER ESCAPE SEQUENCE RANGE TABLE *
1767     4E41      . . .      ;*****
1768     4E3E      . . .      STJTAB EQU $-3
1769     4E41      41 48 .      DB 101Q,110Q ;UPPER CASE <A> - <H>
1770     4E43      17 CE .      DW STJ120+B15 ;SET APPROPRIATE PARAMETER
1771     4E45      . . .      ;
1772     4E45      4A 4E .      DB 112Q,116Q ;UPPER CASE <J> - <N>
1773     4E47      16 CE .      DW STJ110+B15 ;SET APPROPRIATE PARAMETER
1774     4E49      . . .      ;
1775     4E49      50 5A .      DB 120Q,132Q ;UPPER CASE <P> - <Z>
1776     4E4B      15 CE .      DW STJ100+B15 ;SET APPROPRIATE PARAMETER
1777     4E4D      . . .      ;
1778     4E4D      61 68 .      DB 141Q,150Q ;LOWER CASE <A> - <H>
1779     4E4F      17 CE .      DW STJ120+B15 ;SET APPROPRIATE PARAMETER
1780     4E51      . . .      ;
1781     4E51      6A 6E .      DB 152Q,156Q ;LOWER CASE <J> - <N>
1782     4E53      16 CE .      DW STJ110+B15 ;SET APPROPRIATE PARAMETER
1783     4E55      . . .      ;
1784     4E55      70 7A .      DB 160Q,172Q ;LOWER CASE <P> - <Z>
1785     4E57      15 CE .      DW STJ100+B15 ;SET APPROPRIATE PARAMETER
1786     4E59      . . .      ;*****
1787     4E59      . . .      ; STKTAB - SET LATCHING KEYS ESC SEQ RANGE TABLE *
1788     4E59      . . .      ;*****
1789     4E56      . . .      STKTAB EQU $-3
1790     4E59      20 20 .      DB 40Q,40Q ;SPACE
1791     4E5B      0F CE .      DW STJMP3+B15 ;IGNORE
1792     4E5D      . . .      ;
1793     4E5D      30 31 .      DB 60Q,61Q ;NUMBERS <0> AND <1>
1794     4E5F      46 80 .      DW ZDCNUM+B15 ;ACCUMULATE INPUT VALUE
1795     4E61      . . .      ;
1796     4E61      41 43 .      DB 101Q,103Q ;UPPER CASE <A> - <C>
1797     4E63      86 CE .      DW STK020+B15 ;SET APPROPRIATE PARAMETER
1798     4E65      . . .      ;
1799     4E65      52 52 .      DB 122Q,122Q ;UPPER CASE <R>
1800     4E67      81 CE .      DW STK010+B15 ;SET REMOTE FLAG
1801     4E69      . . .      ;
1802     4E69      61 63 .      DB 141Q,143Q ;LOWER CASE <A> - <C>
1803     4E6B      86 CE .      DW STK020+B15 ;SET APPROPRIATE PARAMETER
1804     4E6D      . . .      ;
1805     4E6D      72 72 .      DB 162Q,162Q ;LOWER CASE <R>
1806     4E6F      81 CE .      DW STK010+B15 ;SET REMOTE FLAG
1807     4E71      . . .      ;
1808     4E71      00 7F .      DB 0Q,177Q ;ALL OTHER CODES
1809     4E73      4F 80 .      DW ZESCND+B15 ;ABORT ESCAPE SEQUENCE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 55
1811	4E75	.	.	*****	
1812	4E75	.	.	; STLKYS - SET LATCHING KEYS ESCAPE SEQUENCE *	
1813	4E75	.	.	*****	
1814	4E75	.	.	STLKYS EQU \$	
1815	4E75	3A	F3 FF	LDA MDFLG2 ;INITIALIZE PARAMETER VALUE	
1816	4E78	32	DB FF	STA PARM1	
1817	4E7B	21	56 4E	LXI H,STKTAB ;PUT RANGE TABLE ADDR IN H,L	
1818	4E7E	C3	01 4E	JMP STJMP1 ;EXIT TO WAIT LOOP	
1819	4E81	.	.	*****	
1820	4E81	.	.	; REMOTE FLAG PARAMETER - SET BIT VALUE *	
1821	4E81	.	.	*****	
1822	4E81	.	.	STK010 EQU \$	
1823	4E81	3E	03 .	MVI A,REMBIT ;PUT BIT NUMBER IN A-REGISTE	
1824	4E83	C3	89 4E	JMP STK050 ;SET THE BIT VALUE	
1825	4E86	.	.	*****	
1826	4E86	.	.	; INPUT PARAMETER RECEIVED - SET PARAMETER MASK *	
1827	4E86	.	.	*****	
1828	4E86	.	.	STK020 EQU \$	
1829	4E86	3E	43 .	MVI A,A+2 ;COMPUTE BIT INDEX	
1830	4E88	91	. .	SUB C ;<A>=2, =1, <C>=0	
1831	4E89	.	.	;	
1832	4E89	.	.	STK050 EQU \$	
1833	4E89	21	DB FF	LXI H,PARM1 ;SET HL TO PARAMETER ADDRESS	
1834	4E8C	CD	B7 4E	CALL STPAR1 ;SET PARAMETER MASK	
1835	4E8F	79	. .	MOV A,C ;RECALL INPUT CHARACTER	
1836	4E90	E6	20 .	ANI 40Q ;UPPER CASE CHARACTER?	
1837	4E92	C2	09 4E	JNZ STJMP2 ;NO - CLEAR ACCUMULATOR AND	
1838	4E95	.	.	;	
1839	4E95	11	DB FF	LXI D,PARM1 ;YES - SET LATCHING KEYS	
1840	4E98	21	F3 FF	LXI H,MDFLG2 ;(D,E=SOURCE; H,L=DEST)	
1841	4E9B	1A	. .	LDAX D ;GET INPUT PARAMETER	
1842	4E9C	3A	11 FF	LDA KBFLGS ;EXTRACT INHIBIT MASK	
1843	4E9F	E6	02 .	ANI PERMBM ;(PERMANENT BLOCK MODE)	
1844	4EA1	.	.	;	
1845	4EA1	.	.	;	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 56
=====
1847     4EA1      . . .      ;*****
1848     4EA1      . . .      ; STBITS - SET FLAG BITS FROM PARAMETER MASK *
1849     4EA1      . . .      ;*****
1850     4EA1      . . .      ;
1851     4EA1      . . .      ; ENTRY    A = CHANGE INHIBIT MASK
1852     4EA1      . . .      ;          BIT SET TO 1 TO INHIBIT CHANGES
1853     4EA1      . . .      ;          D,E = INPUT PARAMETER
1854     4EA1      . . .      ;          H,L = WORD TO BE SET
1855     4EA1      . . .      ;
1856     4EA1      . . .      ; EXIT     'A,B DESTROYED
1857     4EA1      . . .      ;          (H,L) UPDATED
1858     4EA1      . . .      ;
1859     4EA1      . . .      STBITS EQU $
1860     4EA1      2F . .      CMA          ;INVERT INHIBIT MASK
1861     4EA2      47 . .      MOV   B,A    ;SAVE INHIBIT MASK
1862     4EA3      1A . .      LDAX D      ;GET PARAMETER VALUE
1863     4EA4      AE . .      XRA  M      ;EXTRACT CHANGES TO SETTINGS
1864     4EA5      A0 . .      ANA  B      ;MASK OUT INHIBITED CHANGES
1865     4EA6      AE . .      XRA  M      ;ALTER APPROPRIATE BITS
1866     4EA7      77 . .      MOV  M,A    ;UPDATE CURRENT SETTINGS
1867     4EA8      1B . .      DCX  D      ;INCREMENT TO NEXT VALUES
1868     4EA9      2B . .      DCX  H
1869     4EAA      C9 . .      RET          ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS
1871	4EAB	.	.	;*****
1872	4EAB	.	.	; STPARM - SET PARAMETER MASK *
1873	4EAB	.	.	;*****
1874	4EAB	.	.	;
1875	4EAB	.	.	; ENTRY A = BIT/WORD INDEX
1876	4EAB	.	.	; H = BASEH
1877	4EAB	.	.	; IODATA = PARAMETER VALUE
1878	4EAB	.	.	;
1879	4EAB	.	.	; EXIT PARAMETER WORD MASK SET
1880	4EAB	.	.	; A,B,L DESTROYED
1881	4EAB	.	.	;
1882	4EAB	.	.	STPARM EQU \$
1883	4EAB	47	.	MOV B,A ;SAVE INDEX IN B-REGISTER
1884	4EAC	E6	F8	ANI 370Q ;MASK OUT 3 LEASE SIGNIFICAN
1885	4EAE	0F	.	RRC ;BITS AND DIVIDE BY 8 TO
1886	4EAF	0F	.	RRC ;DETERMINE WORD INDEX (0-2
1887	4EB0	0F	.	RRC
1888	4EB1	2F	.	CMA ;SET TO NEGATIVE INDEX
1889	4EB2	3C	.	INR A
1890	4EB3	C6	DB	ADI PARM1-CMSTOR ;COMPUTE JUMPER WORD ADD
1891	4EB5	6F	.	MOV L,A ;PUT LSB INTO L (H = BASEH)
1892	4EB6	.	.	;*****
1893	4EB6	.	.	; DETERMINE BIT TO BE MODIFIED *
1894	4EB6	.	.	;*****
1895	4EB6	78	.	MOV A,B ;RECALL CHARACTER INDEX
1896	4EB7	.	.	STPAR1 EQU \$
1897	4EB7	E6	07	ANI 70 ;EXTRACT BIT NUMBER
1898	4EB9	47	.	MOV B,A
1899	4EBA	3E	80	MVI A,200Q ;SET INITIAL BIT LOCATION
1900	4EBC	.	.	;
1901	4EBC	.	.	STP010 EQU \$
1902	4EBC	07	.	RLC ;ROTATE ONE BIT LEFT
1903	4EBD	05	.	DCR B ;BIT POSITION FOUND?
1904	4EBE	F2	BC	JP STP010 ;NO - CONTINUE ROTATING
1905	4EC1	47	.	MOV B,A ;YES - SAVE BIT MASK IN B
1906	4EC2	.	.	;*****
1907	4EC2	.	.	; DETERMINE WHETHER BIT IS TO BE SET OR CLEARED *
1908	4EC2	.	.	;*****
1909	4EC2	EB	.	XCHG ;SAVE WORD ADDRESS IN D,E
1910	4EC3	2A	DE	LHLD IODATA ;GET PARAMETER VALUE
1911	4EC6	7D	.	MOV A,L
1912	4EC7	84	.	ORA H ;BIT TO BE SET?
1913	4EC8	EB	.	XCHG ;(RESTORE H,L)
1914	4EC9	78	.	MOV A,B ;(RECALL BIT TO MODIFY)
1915	4ECA	CA	D0	JZ STP020 ;NO - SET PARM BIT TO CLEAR
1916	4ECD	86	.	ORA M ;YES - ADD BIT TO PARAMETER
1917	4ECE	77	.	MOV M,A ;SET PARAMETER
1918	4ECF	C9	.	RET ;RETURN

13255

2648A MICROCODE LISTING 'KG14'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 58
=====
1920     4ED0      . . .      ;*****
1921     4ED0      . . .      ; CLEAR BIT IN PARAMETER *
1922     4ED0      . . .      ;*****
1923     4ED0      . . .      STP020 EQU $
1924     4ED0     2F . .      CMA                               ;SET TO CLEAR MASK
1925     4ED1     A6 . .      ANA M                             ;CLEAR BIT FROM PARAMETER
1926     4ED2     77 . .      MOV M,A                           ;SET PARAMETER
1927     4ED3     C9 . .      RFT                               ;RETURN
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1929	4ED4	.	.	*****	59
1930	4ED4	.	.	; ALPCHK - CHECK FOR ALPHA TYPE CHARACTER *	
1931	4ED4	.	.	*****	
1932	4ED4	.	.	ALPCHK EQU \$	
1933	4ED4	FE	20	CPI ABLNK ;IS CHARACTER A BLANK?	
1934	4ED6	C8	.	RZ ;YES - RETURN OK (Z TRUE)	
1935	4ED7	E6	DF	ANI 377Q-40Q ;NO - FORCE TO UPPER CASE	
1936	4ED9	FE	41	CPI A ;IS IT ABOVE LETTER A?	
1937	4EDB	F8	.	RM ;NO - RETURN FAIL (Z FALSE)	
1938	4EDC	FE	5A	CPI Z ;IS IT Z OR ABOVE	
1939	4EDE	F0	.	RP ;YES - RETURN (Z TRUE IF =)	
1940	4EDF	BF	.	CMP A ;NO - SET Z TRUE	
1941	4EE0	C9	.	RET ;RETURN GOOD (Z TRUE)	
1942	4EE1	.	.	*****	
1943	4EE1	.	.	; NUMCHK - CHECK FOR NUMERIC TYPE CHARACTER *	
1944	4EE1	.	.	*****	
1945	4EE1	.	.	NUMCHK EQU \$	
1946	4EE1	FE	20	CPI ABLNK ;IS CHARACTER A BLANK?	
1947	4EE3	C8	.	RZ ;YES - RETURN OK (Z TRUE)	
1948	4EE4	FE	2B	CPI PLUS ;BELOW PLUS?	
1949	4EE6	D8	.	RC ;YES - RETURN FAIL	
1950	4EE7	FE	2F	CPI SLANT ;SLANT CHARACTER?	
1951	4EE9	CA	76 4D	JZ NZEXIT ;YES - RETURN FAIL	
1952	4EEC	FE	39	CPI ZERO+9 ;NINE OR BELOW?	
1953	4EEE	F0	.	RP ;NO - RETURN (Z TRUE IF =)	
1954	4EEF	BF	.	CMP A ;YES - SET Z TRUE	
1955	4EF0	C9	.	RET ;RETURN	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 60
=====
1957      4EF1      . . .      ;*****
1958      4EF1      . . .      ; THIS ROUTINE WAS MOVED FROM I0271 TO MAKE ROOM
1959      4EF1      . . .      ; FOR PATCHES
1960      4EF1      . . .      ;*****
1961      4EF1      . . .      ;
1962      FF4F      . . .      IOCERR EQU 177517Q ;I/O ERROR FLAG
1963      0055      . . .      U EQU 125Q ;ASCII U
1964      0009      . . .      STOPRP EQU 11Q ;STOP KEY FROM REPEATING
1965      4EF1      . . .      ;
1966      4EF1      . . .      ; * * * * *
1967      4EF1      . . .      ;
1968      4EF1      . . .      ; RETSCN - SEE IF "RETURN" HAS BEEN HIT
1969      4EF1      . . .      ;
1970      4EF1      . . .      ; RETSCO - IF IN "RECORD" MODE, CHECK FOR
1971      4EF1      . . .      ; RECORD KEY INSTEAD.
1972      4EF1      . . .      ;
1973      4EF1      . . .      ; ENTRY
1974      4EF1      . . .      ;
1975      4EF1      . . .      ; EXIT NC,NZ => NO TERMINATOR
1976      4EF1      . . .      ; C,Z => TERMINATOR
1977      4EF1      . . .      ; IOCERR=U
1978      4EF1      . . .      ; A,L DESTROYED
1979      4EF1      . . .      ;
1980      4EF1      . . .      RETSCO EQU $
1981      4EF1      3A F4 FF      LDA MDFLG1 ;IN "RECORD" MODE?
1982      4EF4      E6 40 .      ANI RECORD
1983      4EF6      2E 9D .      MVI L,235Q
1984      4EF8      C2 FD 4E      JNZ RSC005 ;YES - CHECK FOR RECORD KEY
1985      4EF8      . . .      RETSCN EQU $
1986      4EF8      . . .      ;*****
1987      4EF8      2E EF .      MVI L,SFTCR ;NO - CHECK FOR RETURN
1988      4EFD      . . .      ;*****
1989      4EFD      . . .      RSC005 EQU $
1990      4EFD      3A 4F FF      LDA IOCERR ;HAS CR ALREADY BEEN HIT?
1991      4F00      FE 55 .      CPI U ;'U' => YES
1992      4F02      CA 1B 4F      JZ RSC020 ;YES - RETURN
1993      4F05      C5 . . .      PUSH B ;NO - SAVE REGISTERS FOR CAL
1994      4F06      D5 . . .      PUSH D ;TO GETKY
1995      4F07      CD 75 49      CALL GTKEY ;ANY NEW KEYS HIT?
1996      4F0A      D1 . . .      POP D
1997      4F0B      C1 . . .      POP B
1998      4F0C      C0 . . .      RNZ ;NO - RETURN
1999      4F0D      8D . . .      CMP L ;TARGET KEY HIT?
2000      4F0E      C2 FD 4E      JNZ RSC005 ;NO - CHECK FOR MORE INPUT
2001      4F11      3E 09 .      MVI A,STOPRP ;YES - INHIBIT KEY REPEAT
2002      4F13      CD 52 4C      CALL KBCTL
2003      4F16      . . .      USRINT EQU $
2004      4F16      3E 55 .      MVI A,U ;SET "IOCERR" TO "U"
2005      4F18      32 4F FF      STA IOCERR ;IOCERR=U => USER INTERRUPT
2006      4F1B      . . .      RSC020 EQU $ ;RETURN "USER INTERRUPT"
=====

```


13255
2648A MICROCODE LISTING 'KG14'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 61
=====
2007     4F1B     37 . .      STC                                     ;C => ERROR
2008     4F1C     C9 . .      RET
```

13255

2648A MICROCODE LISTING 'KG14'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS          PAGE 62
=====
2010     4F1D      .      .      .      END
          0  ERRORS FOUND IN ASSEMBLY CODE .
```

13255

2648A MICROCODE LISTING 'KG14'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
A          0041      305, 1743, 1829, 1936
ABLNK     0020      303, 1933, 1946
ALKYNM    0020      418, 811
ALPCHK    4ED4     1932, 477
ALTOUT    482A     483
AUTOLF    0004     140, 833, 1087
AUTTRM    0001      71
B15       8000     310, 1770, 1773, 1776, 1779, 1782, 1785, 1791, 1794, 1797,
                1800, 1803, 1806, 1809
BACKTB    00E9     338, 899
BELL      4DD0     1669, 472, 1471
BELLED    0080     273, 1390, 1393, 1412, 1676, 1698
BKSPCE    0008     296, 887
BLKDLY    001E     282, 1136
BLKFLG    FF0E     367, 368, 1137, 1456, 1570, 1612
BLKMDE    0002     139, 827, 1093, 1364
BLKSET    00FF     283
BLKTMR    91ED     382, 383
BLKTRG    0001      95
BLKTRM    5004     208, 209
BMKYNM    0040     419, 808
BRKCOL    0001     412, 1534
BRKYRN    000D     411, 1533
CAPSLK    0001     138, 839, 907, 1081
CHECK1    0001     604, 583, 982
CHECK2    0002     605, 585, 984
CHEKCC    0040      88
CKB010    4D6F     1537, 1540
CKBRK1    4D6F     1536, 1487
CKBRKY    4D64     1529, 1317
CKC010    4B6C     1069, 1067
CKDSFN    4B3C     1042, 612
CKI010    4D55     1505, 1499
CKICHR    4B58     1060, 613
CKIOKY    4D2B     1482, 1315
CKMLOK    4B2E     1032, 611
CLKYNM    0010     417, 813
CLRLNE    00CB     324, 581
CLRMD1    4D9F     1602, 471
CLRTAB    00B2     315, 574
CLRTBS    00B3     316, 575
CLRTRG    0000     232
CLRTRM    0002      72
CLRXTM    4DE7     1695, 474
CLSCRN    00CA     323, 580
CMBASE    00FF     156, 157
CMFLGS    FFF8     165, 166, 998, 1396, 1416, 1467
CMSTOR    FF00     157, 1368, 1396, 1890
COMMON    FFFF     155, 156, 159
CONDIS    0001      42
CONDTN    009F     351, 589
CPSADJ    0020     299, 914
CR         000D     447

```

13255

2648A MICROCODE LISTING 'KG14'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```
=====
```

CTIJMP	FFE0	180, 181
CTIVEC	FFE1	179, 180
CTLKEY	0001	400, 884, 949, 1047, 1072
CTLMSK	001F	298, 893
CTRDKY	00A0	352, 573
CURLFT	00C4	321, 576
CURRHT	00C3	320, 578
DC2SND	0080	61, 1366
DCHWRP	00CF	326, 957, 1012
DCIOFF	0010	119
DCJMP0	0080	76
DCJMP1	0001	80
DCJMP2	0002	81
DCJMP3	0004	82
DCJMP4	0008	83
DCJMS2	5006	210, 1757
DCJMSK	5005	209, 210, 1760
DCMERR	0001	104
DECRDX	000A	146, 1717
DEFSKY	0008	98, 999
DEL	007F	308, 912
DELCHR	00D0	327, 582, 955
DFNCOF	009A	349, 1054
DISCNT	0006	238
DISPST	FFFE	159, 160
DSFLED	0001	266, 1454, 1458, 1650
DSPFNC	0001	127, 1044, 1462
DSPFON	00D9	333, 1048
EDIT	0010	131
EDTLED	0010	270, 1654
EDTWRP	0008	74, 952, 1009, 1066
ENDBLK	0007	239
ENDPRF	00DB	334, 593
ENDTS0	4CD3	1409, 1582, 1616
ENDTST	4CD8	1415, 1313
ENHNCF	00FF	319, 591
ENTCOL	0002	409, 1498
ENTRCD	0098	348, 1253, 1498
ERRFLG	FFF7	166, 167
ESCFLG	FFD1	192, 195, 1725
EXFNLM	00A1	353, 1260
F1FUNC	00F0	341, 590, 592, 594, 596, 598, 600
FLD010	4DBE	1639, 1642
FMTOFF	00D8	332, 599
FMTONF	00D7	331, 597
FNBASE	0080	619, 1019
FNCLIM	0087	620, 946
FNCTAB	4935	610, 1022
FNDLED	4DBA	1635, 1563, 1609
FORGN	0080	134
FORMAT	0008	130
FRCPTY	0080	90
FRCRST	0004	97, 1398, 1418

13255

13255/90010
REV 04/17/78

2648A MICROCODE LISTING 'KG14'

SYMBOL VALUE REFERENCED ON

```

=====
FRSALT 4829 482
FSTBIN 000A 242
FSTRAM 9100 29, 152, 374
FSTSND 0020 56
FULDUP 0080 33
GRAFUN 0097 290, 1257
GTK005 4984 698, 773
GTK010 498E 703, 801, 810, 816, 826, 846, 881, 1107
GTK020 4998 708
GTK025 4987 728, 724, 726
GTK030 4988 731, 742
GTK040 498E 737, 696
GTK060 49C7 746, 718, 795, 874
GTK100 49CB 754, 701, 707
GTK120 49E3 770, 768
GTK130 49EA 777, 764
GTK150 4A0B 807, 798
GTK160 4A2A 823, 809
GTK170 4A37 832, 812
GTK180 4A3C 838, 814
GTK190 4A3E 841, 819, 828, 834
GTK200 4A47 851, 788
GTK210 4A58 868, 863
GTK215 4A8D 898, 888, 890
GTK220 4A92 905, 885
GTK230 4AA7 918, 892, 894, 900, 909, 911, 913
GTK240 4ABB 936, 930
GTK300 4AC1 945, 882
GTK310 4AE1 964, 950
GTK320 4AE7 969, 975
GTK330 4AF4 980, 972
GTK350 4AFF 990, 954, 956, 958, 976, 1001, 1003, 1011, 1013
GTK370 4B05 997, 985
GTK380 4B13 1007, 983
GTK400 4B21 1018, 947
GTK410 4B53 1053, 1045
GTK450 4B8D 1101, 1082, 1088, 1094
GTKEY 4975 691, 467, 1538, 1995
GTKYNM 4C1C 1223, 789, 859
GTKYX1 4AAC 924, 733, 992, 1036, 1038, 1049, 1051, 1055, 1064, 1074,
1076
GTN010 4C23 1234, 1237
HNDSHK 0040 58, 1366
HOMEDN 00C6 322, 587
HOMEUP 00E8 337, 586
ICHLLED 0002 267, 1651
ICHROF 00D2 329, 1063
ICHRON 00D1 328, 1073
INITK0 495B 644, 1433
INITK1 4971 654
INITKB 4943 634, 466
INSCHR 0002 128, 1062
INSWRP 0002 96

```


13255

13255/90010

2648A MICROCODE LISTING 'KG14'

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
KBPTPT  91C2    393, 394, 651, 699, 705, 1190, 1205
KBSBSE  00FF    359, 360
KBSRAM  FF20    358, 359, 361
KBSSTR  FF00    360, 1452, 1456
KBTIMR  91EC    383, 385, 715, 800, 817, 875, 919, 937, 1124, 1251,
          1335, 1466, 1492, 1518
KEYBFL  00C4    392, 766, 769, 1196, 1199
KEYBLN  0028    384, 385, 649, 769, 1199
KEYBUF  91C4    385, 392, 393, 649
KEYCOL  91EE    381, 382, 648, 1149, 1178, 1188
LDRCHK  0004    106
LEDSAV  FF0B    370, 1392, 1420
LEDTAB  4DC8    1649, 1637
LFPOS   0010    50
LINWRP  0004    46
LNGDLY  005B    280, 920, 1252, 1491
LOKKBD  4C79    1333, 1308
LSHFKY  0008    401, 861
LWRASC  4837    499, 487, 862
LWRLIM  0060    307, 910
MDFLG1  FFF4    169, 170, 1033, 1043, 1061, 1460, 1560, 1605, 1981
MDFLG2  FFF3    170, 171, 843, 906, 1102, 1362, 1815, 1840
MEMLOK  0004    129, 1034
MLKLED  0004    268, 1652
MLKON   00EC    340, 1035
MNCDBS  0040    250, 251
MNMDON  00F9    342, 1050
MSGPT1  FFF1    171, 172
MSGPT2  FFEF    172, 173
MSGPT3  FFED    173, 174
MSGPT4  FFEB    174, 175
MSGPT5  FFE9    175, 176
MSGPT6  FFE7    176, 177
MSGPT7  FFE5    177, 178
MSGPT8  FFE3    178, 179
NEXTPG  00D5    330, 584
NODCST  0010    84
NOTEST  0004    73
NRMFCT  4917    571, 603, 966
NUMALT  000F    603, 967
NUMCHK  4EE1    1945, 478
NUMCOL  000D    285, 645, 647, 1161, 1162, 1173, 1174, 1436
NZEXIT  4D76    1541, 1951
OCTRDX  0008    147
PAGSTR  0008    48
PARM1   FFDB    184, 185, 1752, 1816, 1833, 1839, 1890
PARM2   FFDA    185, 186, 1709
PARM3   FFD9    186, 187, 1711
PARM4   FFD8    187, 188
PARM5   FFD7    188, 189
PARM6   FFD5    189, 190
PERMBM  0002    261, 825, 1360, 1843
PLUS    002B    300, 1948

```

13255

2648A MICROCODE LISTING 'KG14'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

=====

POLL	0040	121
PRCCTL	FFF5	168, 169
PRNTAL	0010	75
PROMPT	000D	245
PUTBRK	0005	237
RADIX	FFD4	190, 191, 1718
RCVMDE	0020	100
RDKYCD	009C	350, 572
RECLD	0040	272, 1454, 1458, 1656
RECORD	0040	133, 1462, 1982
RECSEP	5003	207, 208
REMBIT	0003	286, 1823
REMOTE	0008	141, 818, 1099
REMSET	0010	99, 1469
RETSCO	4EF1	1980, 489
RETSCN	4EFB	1985, 488
RMKYNM	0008	416, 815
RNGTA	FFD2	191, 192, 1719
RPTDLY	0006	278, 721
RPTKEY	4CAB	1377, 1310
RPTKY	0008	262, 695, 1337, 1378, 1450
RSC005	4EFD	1989, 1984, 2000
RSC020	4F1B	2006, 1992
RSETDC	0002	234
RSETKB	4CEA	1432, 1314
RSHFKY	0010	402, 861
RSK010	4CF5	1438, 1444
RSTOFF	0004	434
RSTON	0002	433, 1402
RSTTMR	FFD0	195
SCNVEC	9168	153
SELECT	0020	132, 1462
SELLED	0020	271, 1454, 1458, 1655
SETCH	0020	86
SETLCL	0004	236
SETLMG	0084	317, 577
SETMD1	4D78	1557, 470
SETMON	0008	240
SETNRM	0009	241
SETREM	0003	235
SETRMG	0085	318, 579
SETR0M	0000	123
SETRPT	4C2D	1250, 991
SETSFK	00EA	339, 1000
SETTRG	0001	233
SETXMT	4DDE	1685, 473
SFTCR	00EF	446, 1987
SLANT	002F	301, 1950
SNDATN	0008	243
SNDFCT	000C	244
SPLDIS	0002	44
SRTDLY	0033	279, 1272
ST1010	4094	1577, 1572

13255
2648A MICROCODE LISTING 'KG14'

SYMBOL	VALUE	REFERENCED ON
ST1020	4D95	1579, 1575
STAULF	4B81	1086, 615
STBITS	4EA1	1859, 1755, 1758, 1762
STBLKM	4B86	1092, 616
STBLMD	4C92	1357, 1311
STCPLK	4B7C	1080, 614
STJ100	4E15	1736, 1776, 1785
STJ110	4E16	1738, 1773, 1782
STJ120	4E17	1740, 1770, 1779
STJMP1	4E01	1716, 1818
STJMP2	4E09	1720, 1749, 1837
STJMP3	4E0F	1723, 1791
STJMPR	4DF2	1707, 475
STJTAB	4E3E	1768, 1712
STK010	4E81	1822, 1800, 1806
STK020	4E86	1828, 1797, 1803
STK050	4E89	1832, 1824
STKTAB	4E56	1789, 1817
STLKYS	4E75	1814, 476
STOPRP	0009	1964, 2001
STP010	4E8C	1901, 1904
STP020	4ED0	1923, 1915
STPAR1	4EB7	1896, 1834
STPARM	4EAB	1882, 1746
STPRF	00DD	335, 595
STPRPT	4D5E	1516, 1316
STR010	4C4E	1271, 1266
STR020	4C50	1273, 1254, 1258
STRMMD	4B8B	1098, 617
STRTS1	4CCD	1401, 693
STRTST	4CB4	1387, 1312, 1472
STXMOF	00FB	344, 601
TAB	0009	297, 889
TEST	00FA	343, 588, 1496, 1501
TESTOK	0002	105
TMIACK	0000	115
TMIEN	0002	118
TMIOFF	0020	120
TMRINT	0003	110
TMRON	0001	117
TRIGGR	5002	206, 207
TRMTYP	FFFD	160, 161
TSTCOL	0001	405, 1501
TSTROW	FF17	404, 1500
U	0055	1963, 1991, 2004
UNLKBD	4C89	1347, 1309
UPRASC	48A7	535, 864
UPRLIM	0040	304, 891
USRINT	4F16	2003, 490
WBSR	0020	142
XMTLED	0008	269, 1686, 1700
Z	005A	306, 1938
ZCHKSF	00C6	444

13255

2648A MICROCODE LISTING 'KG14'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```
=====
```

ZDCBAS	5000	205, 206, 216
ZDCCTL	5011	219, 220
ZDCINT	5026	226
ZDCMNS	004C	255, 256
ZDCMON	500E	218, 219
ZDCNUM	0046	253, 254, 1794
ZDCPLS	0049	254, 255
ZDCTST	5014	220, 221
ZDSPMS	0040	251, 252
ZERO	0030	302, 1952
ZESCND	004F	256, 1809
ZGCKYS	600E	440, 873
ZGETDC	5017	221, 222
ZGTBIN	501D	223, 224
ZIN2DC	500B	217, 218
ZINIDC	5008	216, 217
ZMDLY	0010	451, 727
ZMOUT	008C	450, 725
ZMUCHK	6023	445
ZNDBIN	5023	225, 226
ZOOMIN	008B	449, 723
ZPUTDC	501A	222, 223
ZRELGC	6011	441, 794
ZRSTDS	0043	252, 253
ZSTBIN	5020	224, 225
ZSTRTS	0005	1327
ZTINT	6014	442, 1122
ZTKCLR	603E	443, 1672
ZTKSTR	6059	448, 1763

395 SYMBOLS, 1009 REFERENCES, 25 WORK TRACKS

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
1	0000	. . .	ASB,HEX ;DC16 24MAY77	1
2	0000	. . .	*****	
3	0000	. . .	; THIS IS DC14F MODIFIED FOR GRAPHICS	
4	0000	. . .	; THE MODIFICATIONS ARE	
5	0000	. . .	; 1. IF EITHER STRAPS P OR Q ARE OUT ON POWER UP,	
6	0000	. . .	; (TEK MODE), A VERY LARGE DATACOM BUFFER IS	
7	0000	. . .	; ALLOCATED (2 K).	
8	0000	. . .	; 2. IF IN TEK MODE, NO ENQ-ACK HANDSHAKE IS DONE	
9	0000	. . .	; 3. IF IN TEK MODE, DELS ARENT STRIPPED OUT	
10	0000	. . .	;	
11	0000	. . .	; SYMBOLS USED BY GRAPHICS PATCHES	
12	0000	. . .	;	
13	90AD	. . .	ZTKFLG EQU 1102550 ;TEK MODE FLAGS	
14	0040	. . .	SCLD EQU 1000 ;SCALED MODE ON	
15	0001	. . .	UNSCLD EQU 10 ;UNSCALED MODE ON	
16	0020	. . .	PJMPR EQU 400 ;SCALED TEK MODE ON	
17	0040	. . .	QJMPR EQU 1000 ;UNSCALED TEK MODE ON	
18	0060	. . .	NRMBUF EQU 96 ;NORMAL BUFFER SIZE	
19	0800	. . .	BIGBUF EQU 2048 ;TEK MODE BUFFER SIZE	
20	0000	. . .	;	
21	0000	. . .	*****	
22	0000	. . .	**** THIS IS THE ROM VERSION *****	
23	0000	. . .	;	
24	0000	. . .	; COMMON EQUATES - CM34 - 6/10/76 - 1315 HRS.	
25	0000	. . .	;	
26	9100	. . .	FSTRAM EQU 1104000 ;FAST RAM LOWER LIMIT	
27	0000	. . .	*****	
28	0000	. . .	; KBDCSW - KEYBOARD DATA COMM SWITCHES *	
29	0000	. . .	*****	
30	0080	. . .	FULDUP EQU 2000 ;HALF/FULL DUPLEX	
31	0000	. . .	*****	
32	0000	. . .	; KBJMPR - KEYBOARD INTERFACE JUMPERS *	
33	0000	. . .	*****	
34	0000	. . .	;	
35	0000	. . .	; JUMPERS SENSED AS 0' WHEN INSERTED	
36	0000	. . .	;	
37	0000	. . .	; ALL JUMPERS ARE NORMALLY INSERTED	
38	0000	. . .	;	
39	0001	. . .	CONDIS EQU 0010 ;CONTROL CODE DISABLE	
40	0000	. . .	; (0=DISABLED)	
41	0002	. . .	SPLDIS EQU 0020 ;SPOW LATCH DISABLE	
42	0000	. . .	; (0=DISABLED)	
43	0004	. . .	LINWRP EQU 0040 ;COLUMN 80 AUTO CR,LF	
44	0000	. . .	; (0=ENABLED)	
45	0008	. . .	PAGSTR EQU 0100 ;PAGE MODE STRAP	
46	0000	. . .	; (0=LINE-FIELD MODE)	
47	0010	. . .	LFPOS EQU 200 ;LINE FEED POSITION	
48	0000	. . .	; (0 = POSITION LINE FEED	
49	0000	. . .	; AT START OF NEXT I/O	
50	0000	. . .	; READ	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  2
=====
51      0000      . . .      ;                                1 = PUT LINE FEED AT END
52      0000      . . .      ;                                OF RECORD)
53      0020      . . .      FSTSND EQU 40Q      ;9600 BAUD DATACOM SHIFT
54      0000      . . .      ;                                (0=9600 BAUD FOR ESC,E)
55      0040      . . .      HNDSHK EQU 100Q    ;BLOCK TRANSFER HANDSHAKE
56      0000      . . .      ;                                (0 = FOLLOW DC2SND SETTING
57      0000      . . .      ;                                1 = SEND DC2 BEFORE DATA)
58      0080      . . .      DC2SND EQU 200Q
59      0000      . . .      ;                                (0 = SEND DC2 ON ENTER
60      0000      . . .      ;                                AND FUNCTION KEY IN
61      0000      . . .      ;                                BLOCK MODE
62      0000      . . .      ;                                1 = INHIBIT ALL DC2
63      0000      . . .      ;                                HANDSHAKE)
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
65	0000	. . .	;*****	
66	0000	. . .	; KBJMP2 - SECOND SET OF KEYBOARD JUMPERS *	
67	0000	. . .	;*****	
68	0001	. . .	AUTTRM EQU 10 ;AUTO TERMINATE ON "ENTER"	
69	0002	. . .	CLRTRM EQU 20 ;CLEAR TERMINATOR ON TRANSMI	
70	0004	. . .	NOTEST EQU 40 ;INHIBIT TERMINAL SELF-TEST	
71	0008	. . .	EDTWRP EQU 100 ;INVERT SENSE OF EDIT WRAP	
72	0010	. . .	PKNTAL EQU 200 ;SEND ALL CODES TO PRINTER	
73	0080	. . .	DCJMP0 EQU 2000 ;DATA CUMM JUMPER	
74	0000	. . .	;*****	
75	0000	. . .	; KBJMP3 - THIRD SET OF KEYBOARD JUMPERS *	
76	0000	. . .	;*****	
77	0001	. . .	DCJMP1 EQU 10 ;DATA CUMM JUMPERS	
78	0002	. . .	DCJMP2 EQU 20 ;.	
79	0004	. . .	DCJMP3 EQU 40 ;.	
80	0008	. . .	DCJMP4 EQU 100 ;.	
81	0010	. . .	NODCST EQU 200 ;INHIBIT DATA CUMM SELF-TEST	
82	0000	. . .	; (0 = DISABLED)	
83	0020	. . .	SETCH EQU 400 ;TURN ON "CH" CONTROL LINE	
84	0000	. . .	; (0 = OFF, 1 = ON)	
85	0040	. . .	CHEKCC EQU 1000 ;MONITOR CC CONTROL LINE	
86	0000	. . .	; (1 = ENABLED)	
87	0080	. . .	FRCPY EQU 2000 ;FORCE PARITY ON/NO IN CHECK	
88	0000	. . .	; (1 = ENABLED)	
89	0000	. . .	;*****	
90	0000	. . .	; CMFLGS - COMMON FLAGS *	
91	0000	. . .	;*****	
92	0001	. . .	BLKTRG EQU 10 ;BLOCK TRANSFER TRIGGER	
93	0002	. . .	INSWRP EQU 20 ;INSERT WITH WRAP AROUND	
94	0004	. . .	FRCKST EQU 40 ;FORCE FULL TERMINAL RESET	
95	0008	. . .	DEFSKY EQU 100 ;DEFINE SOFT KEY MODE ENABLE	
96	0010	. . .	REMSET EQU 200 ;REMOTE MODE ENABLED	
97	0020	. . .	RCVMDE EQU 400 ;TERMINAL IN RECEIVE MODE	
98	0000	. . .	;*****	
99	0000	. . .	; ERRFLG - ERROR FLAGS *	
100	0000	. . .	;*****	
101	0001	. . .	DCMERR EQU 10 ;DATACOM (1 = ERROR)	
102	0002	. . .	TESTOK EQU 20 ;SELF-TEST (0 = ERROR)	
103	0004	. . .	LDRCHK EQU 40 ;LOADER CHECKSUM (0 = ERROR)	
104	0000	. . .	;*****	
105	0000	. . .	; INTFLG - INTERRUPT FLAG *	
106	0000	. . .	;*****	
107	0003	. . .	TMRINT EQU 3 ;TIMER INTERRUPT	

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	4
109	0000	. . .	;*****		
110	0000	. . .	; PRCCTL - PROCESSOR CONTROL FLAGS *		
111	0000	. . .	;*****		
112	0000	. . .	TMIACK EQU 00 ;ACKNOWLEDGE TIMER INTERRUPT		
113	0000	. . .	; (BIT 1 OFF)		
114	0001	. . .	TMRON EQU 10 ;SET TIMER ON		
115	0002	. . .	TMIEN EQU 20 ;RE-ENABLE TIMER INTERRUPT		
116	0010	. . .	DCIOFF EQU 200 ;DISABLE DATA COMM INTERRUPT		
117	0020	. . .	TMI0FF EQU 400 ;DISABLE TIMER INTERRUPTS		
118	0040	. . .	POLL EQU 1000 ;POLL CTU INTERRUPTS		
119	0000	. . .	;V*V*V*V* SET TO ZERO FOR ROM VERSION *V*V*V*V*		
120	0000	. . .	SETROM EQU 0 ;0 = ENABLE ROM		
121	0000	. . .	;*****		
122	0000	. . .	; MDFLG1 - TERMINAL MODE FLAGS 1 *		
123	0000	. . .	;*****		
124	0001	. . .	DSPFNC EQU 10 ;DISPLAY FUNCTIONS ENABLED		
125	0002	. . .	INSCHR EQU 20 ;INSERT CHARACTER ENABLED		
126	0004	. . .	MEMLOK EQU 40 ;MEMORY LOCK ENABLED		
127	0008	. . .	FORMAT EQU 100 ;FORMAT MODE ENABLED		
128	0010	. . .	EDIT EQU 200 ;EDIT MODE ENABLED		
129	0020	. . .	SELECT EQU 400 ;SELECT MODE ENABLED		
130	0040	. . .	RECORD EQU 1000 ;RECORD MODE ENABLED		
131	0080	. . .	FORGN EQU 2000 ;FOREIGN MODE ENABLED		
132	0000	. . .	;*****		
133	0000	. . .	; MDFLG2 - TERMINAL MODE FLAGS 2 *		
134	0000	. . .	;*****		
135	0001	. . .	CAPSLK EQU 10 ;CAPS LOCK ENABLED		
136	0002	. . .	BLKMDE EQU 20 ;BLOCK MODE ENABLED		
137	0004	. . .	AUTOLF EQU 40 ;AUTO LF ENABLED		
138	0008	. . .	REMOTE EQU 100 ;REMOTE ENABLED		
139	0020	. . .	WBSK EQU 400 ;WRITE-BACKSPACE-READ MODE		
140	0000	. . .	;*****		
141	0000	. . .	; RADIX - BASE OF INPUT PARAMETER FOR ESC SEQ *		
142	0000	. . .	;*****		
143	000A	. . .	DECRDX EQU 10 ;DECIMAL NUMBERS		
144	0008	. . .	OCTRDX EQU 8 ;OCTAL NUMBERS		

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE   5
=====
146      0000      . . .      ;*****
147      0000      . . .      ; COMMON VARIABLES *
148      0000      . . .      ;*****
149      9165      . . .      INTVEC EQU  FSTRAM+1450 ;CENTRAL INTERRUPT VECTOR
150      9168      . . .      SCNVEC EQU  INTVEC+3   ;FOREIGN TERMINAL DISPLY SCA
151      0000      . . .      ;
152      FFFF      . . .      COMMON EQU   1777770   ;UPPER LIMIT OF COMMON AREA
153      00FF      . . .      CMBASE EQU  COMMON/256 ;MSB OF COMMON ADDRESSES
154      FF00      . . .      CMSTOR EQU  CMBASE*256 ;MSB ADJUSTMENT FACTOR
155      0000      . . .      ;
156      FFFE      . . .      DISPST EQU  COMMON-1  ;DISPLAY REFRESH START PTR
157      FFFD      . . .      TRMTYP EQU   DISPST-1 ;TERMINAL TYPE NUMBER
158      FFFC      . . .      KBDCSW EQU  TRMTYP-1  ;KEYBOARD DATACOM SWITCHES
159      FFFB      . . .      KBJMPR EQU  KBJMPR-1  ;KEYBOARD STRAPS
160      FFFA      . . .      KBJMP2 EQU  KBJMPR-1  ;SET 2
161      FFF9      . . .      KBJMP3 EQU  KBJMP2-1  ;SET 3
162      FFF8      . . .      CMFLGS EQU  KBJMP3-1  ;COMMON FLAGS
163      FFF7      . . .      ERRFLG EQU  CMFLGS-1  ;ERROR FLAGS
164      FFF6      . . .      INTFLG EQU  ERRFLG-1  ;INTERRUPT FLAG
165      FFF5      . . .      PRCCTL EQU  INTFLG-1  ;PROCESSOR CONTROL FLAGS
166      FFF4      . . .      MDFLG1 EQU  PRCCTL-1  ;TERMINAL MODE FLAGS 1
167      FFF3      . . .      MDFLG2 EQU  MDFLG1-1  ;AND 2
168      FFF1      . . .      MSGPT1 EQU  MDFLG2-2  ;MESSAGE POINTERS
169      FFEF      . . .      MSGPT2 EQU  MSGPT1-2  ;.
170      FFED      . . .      MSGPT3 EQU  MSGPT2-2  ;.
171      FFE8      . . .      MSGPT4 EQU  MSGPT3-2  ;.
172      FFE9      . . .      MSGPT5 EQU  MSGPT4-2  ;.
173      FFE7      . . .      MSGPT6 EQU  MSGPT5-2  ;.
174      FFE5      . . .      MSGPT7 EQU  MSGPT6-2  ;.
175      FFE3      . . .      MSGPT8 EQU  MSGPT7-2  ;.
176      FFE1      . . .      CTIVEC EQU  MSGPT8-2  ;CTU INTERRUPT VECTOR
177      FFE0      . . .      CTIJMP EQU  CTIVEC-1  ;JUMP CODE FOR VECTOR
178      FFDE      . . .      IODATA EQU  CTIJMP-2  ;ESQ SEQ PARM ACCUMULATOR
179      FFDD      . . .      IOCSGN EQU  IODATA-1  ;SIGN FOR PARAMETER
180      FFDC      . . .      IOPSGN EQU  IOCSGN-1  ;PARAMETER SIGN
181      FFDB      . . .      PARM1 EQU  IOPSGN-1  ;ESCAPE SEQUENCE PARAMETERS
182      FFDA      . . .      PARM2 EQU  PARM1-1   ;.
183      FFD9      . . .      PARM3 EQU  PARM2-1   ;.
184      FFD8      . . .      PARM4 EQU  PARM3-1   ;.
185      FFD7      . . .      PARM5 EQU  PARM4-1   ;.
186      FFD5      . . .      PARM6 EQU  PARM5-2   ;.
187      FFD4      . . .      RADIX EQU  PARM6-1   ;RADIX OF PARAMETERS
188      FFD2      . . .      RNGTA EQU  RADIX-2   ;CHAR FUNCTION TABLE ADDRESS
189      FFD1      . . .      ESCFLG EQU  RNGTA-1  ;ESCAPE SEQUENCE FLAG
190      0000      . . .      ;
191      0000      . . .      ;
192      FFD0      . . .      RSTTMR EQU  ESCFLG-1 ;SOFT RESET TIMER
193      0000      . . .      ; * * * * *
194      0000      . . .      ; END OF COMMON EQUATES
195      0000      . . .      ;*****
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE   6
=====
197      0000      . . .      ;*****
198      0000      . . .      ; DCSTAT - DATACOM STATUS BITS *
199      0000      . . .      ;*****
200      0001      . . .      DCDP   EQU 10      ;DATA PRESENT
201      0002      . . .      DCTBE  EQU 0020     ;TRANSMIT BUFFER EMPTY
202      0004      . . .      DCOE   EQU 0040     ;OVERRUN ERROR
203      0008      . . .      DCPE   EQU 0100     ;PARITY ERROR
204      0000      . . .      ;
205      0000      . . .      ;
206      0000      . . .      ; THE FOLLOWING ARE INTERPRETED AS 0 = ON AND
207      0000      . . .      ;   1 = OFF
208      0000      . . .      ;
209      0010      . . .      DCCF   EQU 200      ;CF - RECEIVED CARRIER
210      0020      . . .      DCCB   EQU 400      ;CB - CLEAR TO SEND
211      0040      . . .      DCSB   EQU 100Q     ;SB - SECONDARY RECEIVED DAT
212      0000      . . .      ;
213      0080      . . .      DCCC   EQU 200Q     ;CC - DATA SET READY?
214      0000      . . .      ;
215      0000      . . .      ;*****
216      0000      . . .      ; DCFLGS - DATACOM FLAGS *
217      0000      . . .      ;*****
218      0001      . . .      DCCA   EQU 1        ;REQUEST TO SEND (0 = SET)
219      0002      . . .      TRNMOD EQU 2        ;TRANSPARENT MODE
220      0004      . . .      BINMOD EQU 4        ;BINARY MODE
221      0008      . . .      GOBIN  EQU 100      ;GO TO BINARY MODE ON ENQ-AC
222      0010      . . .      FB9600 EQU 200      ;FAST BINARY MODE
223      0020      . . .      MCMOD  EQU 400      ;MAIN CHANNEL (0 = ENABLED)
224      0040      . . .      SPECHO EQU 100Q     ;ECHO SUPPRESS (1 = ENABLED)
225      0080      . . .      FORPAR EQU 200Q     ;FORCE PARITY MODE
=====

```


ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE	7
227	0000	. . .	;*****		
228	0000	. . .	; I/O MODULE EQUATES *		
229	0000	. . .	;*****		
230	0080	. . .	IOBASE EQU 2000 ;I/O ADDRESS MSB'S		
231	0070	. . .	PROCSK EQU 1600 ;PROCESSOR PORT		
232	0000	. . .	;*****		
233	0000	. . .	; KEYBOARD *		
234	0000	. . .	;*****		
235	8300	. . .	IOKB EQU 3Q+IOBASE*256 ;MODULE 11 BASE ADDRESS		
236	8300	. . .	IOKBLD EQU IOKB+00 ;SET KEYBOARD LED'S		
237	0000	. . .	;*****		
238	0000	. . .	; DATACOM *		
239	0000	. . .	;*****		
240	8100	. . .	IODC EQU 1Q+IOBASE*256 ;MODULE 10 BASE ADDRESS		
241	0000	. . .	;		
242	0000	. . .	; INPUT ADDRESSES		
243	0000	. . .	;		
244	8100	. . .	IODCDI EQU IODC+0Q ;DATACOM DATA IN		
245	8120	. . .	IODCST EQU IODC+40Q ;DATACOM STATUS IN		
246	8121	. . .	IODCS2 EQU IODC+41Q ;STATUS W/TBE = THRE AND TRE		
247	8140	. . .	IODCPC EQU IODC+100Q ;DATACOM STRAPS IN		
248	8160	. . .	DCCTL2 EQU IODC+140Q ;ALTERNATE DATACOM CONTROL		
249	0000	. . .	;		
250	0000	. . .	; OUTPUT ADDRESSES		
251	0000	. . .	;		
252	8140	. . .	IODCCT EQU IODC+100Q ;DATACOM CONTROL OUT		
253	8160	. . .	IODCDO EQU IODC+140Q ;DATACOM DATA OUT		
254	0000	. . .	;		
255	0000	. . .	; DATACOM CONTROL BITS		
256	0000	. . .	;		
257	000E	. . .	B9600 EQU 16Q ;9600 BAUD		
258	0040	. . .	DCSA EQU 100Q ;TURN ON SECONDARY TRANSMIT		
259	0000	. . .	DCOP EQU 0 ;ODD PARITY		
260	0010	. . .	DCEP EQU 20Q ;EVEN PARITY		
261	0020	. . .	DCNP EQU 40Q ;NO PARITY		
262	0000	. . .	;		
263	0008	. . .	CDOFF EQU 10Q ;TURN OFF CD (USE W/DCCTL2)		
264	0080	. . .	DCCH EQU 200Q ;CH BIT IN CONTROL		
265	003E	. . .	BAUDPT EQU 076Q ;BAUD RATE/PARITY MASK		

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE   8
=====
267      0000      . . .      ;
268      0000      . . .      ; KEYBOARD ENTRY VECTOR POINTERS
269      0000      . . .      ;
270      4800      . . .      ZKBBAS EQU 44000Q      ;KEYBOARD START ADDRESS
271      4802      . . .      ZINIKB EQU ZKBBAS+2    ;INITIALIZE KEYBOARD
272      4805      . . .      ZGETKY EQU ZINIKB+3    ;GET KEYBOARD KEY
273      4808      . . .      ZKBCTL EQU ZGETKY+3    ;PERFORM KEYBOARD CONTROL
274      480B      . . .      ZKBMON EQU ZKBCTL+3    ;MONITOR KEYBOARD
275      480E      . . .      ZSTMD1 EQU ZKBMON+3    ;SET MODE 1 FLAGS
276      4811      . . .      ZCLMD1 EQU ZSTMD1+3    ;CLEAR MODE 1 FLAGS
277      4814      . . .      ZBELL EQU ZCLMD1+3     ;SOUND THE BELL
278      4817      . . .      ZSTXMT EQU ZBELL+3     ;SET TRANSMIT LED
279      481A      . . .      ZCLXMT EQU ZSTXMT+3    ;CLEAR TRANSMIT LED
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
281	0000	. . .	;*****	9
282	0000	. . .	; MISCELLANEOUS EQUATES *	
283	0000	. . .	;*****	
284	00C3	. . .	JMP EQU 3030 ;"JMP" CODE	
285	0000	. . .	;*****	
286	0000	. . .	; ASCII CHARACTER EQUATES *	
287	0000	. . .	;*****	
288	0000	. . .	ANULL EQU 00 ;NULL	
289	0001	. . .	SOH EQU 10 ;START OF HEADER	
290	0002	. . .	STX EQU 20 ;START OF TEXT	
291	0003	. . .	ETX EQU 30 ;END OF TEXT	
292	0004	. . .	EOT EQU 40 ;END OF TRANSMISSION	
293	000A	. . .	LF EQU 120 ;LINE FEED	
294	000D	. . .	CR EQU 150 ;CARRIAGE RETURN	
295	0005	. . .	ENQ EQU 50	
296	0006	. . .	ACK EQU 60	
297	0012	. . .	DC2 EQU 220 ;DEVICE CNTL 2	
298	0018	. . .	ACAN EQU 300 ;CANCEL LINE	
299	001E	. . .	RS EQU 360 ;RECORD SEPARATOR	
300	007F	. . .	ADEL EQU 1770 ;DELETE (RUBOUT)	
301	0000	. . .	;*****	
302	0000	. . .	; DISPLAY CONTROL EQUATES *	
303	0000	. . .	;*****	
304	0080	. . .	NORMAL EQU 2000 ;START NORMAL VIDEO	
305	0082	. . .	INVR5 EQU 2020 ;START INVERSE VIDEO	
306	00CC	. . .	EOL EQU 3140 ;END OF LINE FLAG	
307	00CE	. . .	EOP EQU 3160 ;END OF PAGE FLAG	
308	0000	. . .	;*****	
309	0000	. . .	; DATACOM LOCAL VARIABLES *	
310	0000	. . .	;*****	
311	91C0	. . .	DCSTOR EQU FSTRAM+3000	
312	0091	. . .	DCBASE EQU DCSTOR/256	
313	91BF	. . .	DCSTAT EQU DCSTOR-1 ;DATACOM STATUS BITS	
314	91BD	. . .	DCBPTR EQU DCSTAT-2 ;DATACOM BUFFER UNLOAD PTR	
315	91BB	. . .	DCSPTR EQU DCBPTR-2 ;DATACOM LOAD POINTER	
316	91B9	. . .	DCBFBG EQU DCSPTR-2 ;START ADDR OF DATACOM BUFFE	
317	91B8	. . .	DCFLGS EQU DCBFBG-1 ;CONTAINS FOLLOWING FLAGS	
318	91B6	. . .	DCMVEC EQU DCFLGS-2 ;DATACOM MONITOR VECTOR	
319	91B5	. . .	DCMJMP EQU DCMVEC-1 ;JUMP CODE FOR VECTOR	
320	91B4	. . .	XMTDLY EQU DCMJMP-1 ;LIMIT FOR XMIT TURNAROUND	
321	91B3	. . .	DCDLAY EQU XMTDLY-1 ;DELAY FOR SIGNAL SETTling	
322	91B1	. . .	DCTEX EQU DCDLAY-2 ;TURNAROUND EXIT ADDRESS	
323	91B0	. . .	TPARIT EQU DCTEX-1 ;SELF TEST, PARITY	
324	91AF	. . .	TMOCNT EQU TPARIT-1 ;SELF TEST, TIME OUT COUNTER	
325	91AE	. . .	FPMASK EQU TMOCNT-1 ;FORCE PARITY MASK	
326	91AD	. . .	ENDCHR EQU FPMASK-1 ;END OF DATA CHAR FOR MC	
327	91AC	. . .	DCCT EQU ENDCHR-1 ;CONTROL WD FOR CH,CA,NP	
328	0000	. . .	;*****	
329	91AA	. . .	DCBEND EQU DCCT-2 ;ADDRESS OF BUFFER END	
330	0000	. . .	;*****	

13255

2648A MICROCODE LISTING 'DC16'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  10
=====
  331     0080      . . .      GPASYC EQU  200Q      ;GENERAL PURPOSE ASYNC FLAG
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 11
333	0000	. . .	;	
334	0000	. . .	; NEW STRAPPING OPTIONS WHEN 202C MAIN CHANNEL	
335	0000	. . .	; PROTOCOL IS DESIRED. THESE STRAPS ARE	
336	0000	. . .	; AVAILABLE FROM THE NEW KEYBOARD I/F AND	
337	0000	. . .	; ARE SWITCHES R - Z WHICH ARE CURRENTLY	
338	0000	. . .	; NOT USED BY THE FIRMWARE FOR DATACOM.	
339	0000	. . .	;	
340	0080	. . .	SBSTRP EQU 2000 ;(R) 0 = ENABLE CIRCUIT ASSURANCE	
341	0000	. . .	;(SB)	
342	0000	. . .	; 1 = DISABLE CIRCUIT ASSURANCE	
343	0000	. . .	;	
344	0001	. . .	STXSTP EQU 10 ;(S) 0 = USE STX FOR STRT OF DATA	
345	0000	. . .	; 1 = NO START OF DATA CHAR	
346	0000	. . .	;	
347	0002	. . .	ETXSTP EQU 20 ;(T) 0 = USE EOT FOR END OF DATA	
348	0000	. . .	; 1 = USE ETX FOR END OF DATA	
349	0000	. . .	;	
350	0000	. . .	;	
351	0003	. . .	MNCHAN EQU 30 ;00 = NON-MAIN CHANNEL PROTCL	
352	0000	. . .	; 01 = MAIN CHNL W/O SD,ED = EOT	
353	0000	. . .	; 10 = MAIN CHNL W/STX AS SD,	
354	0000	. . .	; 11 = MAIN CHNL W/ETX AS ED	
355	0000	. . .	;	
356	0004	. . .	CBKSTP EQU 40 ;(U) 0 = ENABLE CPU BREAK ON SB	
357	0000	. . .	; 1 = DISABLE CPU BREAK	
358	0000	. . .	;	
359	0008	. . .	CFSTRP EQU 100 ;(V) 0 = ENABLE CF DETECT	
360	0000	. . .	; 1 = DISABLE CF DETECT	
361	0000	. . .	;	
362	0000	. . .	;NODCST EQ 20B (W) 0 = ENABLE DC SELF TEST	
363	0000	. . .	; 1 = DISABLE DC SELF TEST	
364	0000	. . .	;	
365	0000	. . .	;SETCH EQU 40B (X) 0 = SET CH OFF	
366	0000	. . .	; 1 = SET CH ON	
367	0000	. . .	;	
368	0000	. . .	;CHEKCC EQ 100B (Y) 0 = USE CB FOR XMIT LED	
369	0000	. . .	; 1 = USE CC FOR XMIT LED	
370	0000	. . .	;	
371	0000	. . .	;FRCPTY EQ 200B (Z) 0 = DISABLE FORCE PARITY	
372	0000	. . .	; 1 = ENABLE FORCE PARITY	
373	0000	. . .	;	
374	0000	. . .	;	
375	0000	. . .	;	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 12
=====
377      0000      . . .          ORG 50000Q      ;START IN ITS OWN BLOCK
378      5000      . . .          ;
379      5000      . . .          ;*****
380      5000      54 . . .      DB 124Q          ;GRAPHICS VERSION = 'T'
381      5001      . . .          ;*****
382      5001      50 . . .      DB 50000Q/256
383      5002      11 . . .      TRIGGR DB 021Q   ;DC1 FOR TRANSFER TRIGGER
384      5003      1F . . .      RECSEP DB 037Q   ;US FOR RECORD SEPARATOR
385      5004      1E . . .      BLKTRM DB 036Q   ;RS FOR BLOCK TERMINATOR
386      5005      03 . . .      DCJMSK DB 003Q   ;DATA COMM JUMPER ALTER
387      5006      . . .          ;           INHIBIT MASK - SET TO 1
388      5006      . . .          ;           IN APPROPRIATE BIT (0-7)
389      5006      . . .          ;           TO INHIBIT ALTERATION OF
390      5006      . . .          ;           JUMPERS S-Z BY ESCAPE
391      5006      . . .          ;           SEQUENCE
392      5006      . . .          ;           SET TO INHIBIT S,T
393      5006      00 . . .      DCJMK2 DB 000Q   ;STRAP R
394      5007      00 . . .          DB 00Q
395      5008      . . .          ;
396      5008      . . .          ;   DATACOM ENTRY VECTORS
397      5008      . . .          ;
398      5008      C3 7D 51      JMP INITDC      ;INITIALIZE DATACOM
399      5008      C3 91 51      JMP INI2DC      ;INITIALIZATION CONTINUATOR
400      500E      C3 85 91      JMP DCMJMP      ;GO TO MONITOR ROUTINE
401      5011      C3 78 54      JMP DCCTL       ;PERFORM CONTROL FUNCTIONS
402      5014      C3 0B 56      JMP DCTST       ;DATACOM SELF TEST
403      5017      C3 0B 52      JMP GETDC       ;GET A DATACOM CHARACTER
404      501A      C3 9A 50      JMP PUTDC       ;OUTPUT A CHARACTER TO DATAC
405      501D      C3 15 53      JMP GETBIN      ;GET A BINARY BYTE
406      5020      AF . . .      XRA A
407      5021      C9 . . .      RET
408      5022      00 . . .      DB 0Q          ;NOOP START BINARY ROUTINE
409      5023      C3 29 55      JMP TRMBIN      ;TERMINATE BINARY OUTPUT
410      5026      . . .          ;
411      5026      . . .          ;   DATACOM CONTROL CALL CODES
412      5026      . . .          ;
413      0000      . . .          CLRTRG EQU 0    ;CLEAR BLOCK TRANSFER TRIGGE
414      0001      . . .          SETTRG EQU 1    ;SET BLOCK TRANSFER TRIGGER
415      0002      . . .          RSETDC EQU 2    ;RESET DATACOM
416      0003      . . .          SETREM EQU 3    ;SET REMOTE MODE
417      0004      . . .          SETLCL EQU 4    ;SET LOCAL MODE
418      0005      . . .          PUTBRK EQU 5    ;OUTPUT BREAK SIGNAL
419      0006      . . .          DISCNT EQU 6    ;MODEM DISCONNECT
420      0007      . . .          ENDBLK EQU 7    ;SEND ED IF MAIN CHANNEL
421      0008      . . .          SETMON EQU 8    ;ENTER MONITOR MODE
422      0009      . . .          SETNRM EQU 9    ;ENTER NORMAL MODE
423      000D      . . .          PROMPT EQU 13   ;SEND DC2 PROMPT
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 13
425	5026	.	.	;	
426	5026	.	.	;*****	
427	5026	.	.	; DATACOM INTERRUPT PROCESSING *	
428	5026	.	.	;*****	
429	5026	.	.	DCINTR EQU \$	
430	5026	F5	.	PUSH PSW ;SAVE A,FLAGS	
431	5027	C5	.	PUSH B ;SAVE B,C	
432	5028	3A	20 81	LDA IODCST ;GET DATA COMM STATUS	
433	5026	47	.	MOV B,A ;SAVE STATUS IN B	
434	502C	E6	01 .	ANI DCDP ;DATA PRESENT?	
435	502E	CA	82 50	JZ INT170 ;NO - EXIT	
436	5031	E5	.	PUSH H ;YES - SAVE H,L	
437	5032	21	00 81	LXI H,IODCDI ;GET DATA FROM DATA COMM	
438	5035	4E	.	MOV C,M ;INTO C-REGISTER	
439	5036	3A	B8 91	LDA DCFLGS ;CURRENT MODE?	
440	5039	6F	.	MOV L,A ;SAVE DCFLGS	
441	503A	E6	06 .	ANI TRNMOD+BINMOD	
442	503C	C2	59 50	JNZ INT020 ;TRANSPARENT OR BINARY - JUM	
443	503F	79	.	MOV A,C ;ASCII - HANDLE PARITY	
444	5040	E6	7F .	ANI 1770 ;REMOVE PARITY BIT	
445	5042	CA	81 50	JZ INT160 ;RETURN IF NULL	
446	5045	FE	7F .	CPI ADEL	
447	5047	.	.	;*****	
448	5047	.	.	; IF IN TEK MODE, DONT STRIP OUT DEL	
449	5047	CC	94 50	CZ ZCHKTK ;IN TEK MODE? (NZ IF YES)	
450	504A	.	.	;*****	
451	504A	CA	81 50	JZ INT160 ;IGNORE RUBOUTS	
452	504D	78	.	MOV A,B ;GET PARITY BIT	
453	504E	E6	08 .	ANI DCPE ;PARITY ERROR?	
454	5050	CA	59 50	JZ INT020 ;NO JMP	
455	5053	7D	.	MOV A,L ;GET DCFLGS	
456	5054	E6	80 .	ANI FORPAR ;IS FORCE PARITY IN EFFECT?	
457	5056	CA	5F 50	JZ INT050 ;NO - JMP PE ERROR	
458	5059	.	.	INT020 EQU \$;CHECK FOR BUFFER OVERWRITE	
459	5059	78	.	MOV A,B ;RECALL STATUS	
460	505A	E6	04 .	ANI DCOE ;OVERWRITE?	
461	505C	CA	68 50	JZ INT100 ;NO - STORE CHAR	
462	505F	.	.	;*****	
463	505F	.	.	; DATA COMM ERROR - STORE ALL ONES AND SET FLAG *	
464	505F	.	.	;*****	
465	505F	.	.	INT050 EQU \$;DATACOM ERROR	
466	505F	0E	FF .	MVI C,3770 ;STORE ALL ONES	
467	5061	21	F7 FF	LXI H,ERRFLG ;SET ERROR FLAG	
468	5064	7E	.	MOV A,M	
469	5065	F6	01 .	ORI DCMERR	
470	5067	77	.	MOV M,A	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
472	5068	.	.	;*****	14
473	5068	.	.	; STORE CHARACTER IN DATA COMM BUFFER *	
474	5068	.	.	;*****	
475	5068	.	.	INT100 EQU \$	
476	5068	2A	BB 91	LHLD DCSPTR ;GET THE LOAD POINTER	
477	5068	.	.	;*****	
478	5068	.	.	; SEE IF START ADDRESS HAS BEEN REACHED	
479	506B	.	.	; IF SO, SET HL TO END ADDRESS + 1	
480	506B	CD	86 50	CALL INT180	
481	506E	.	.	;*****	
482	506E	.	.	;	
483	506E	.	.	INT130 EQU \$	
484	506E	2B	.	DCX H ;DECREMENT TO NEXT LOCATION	
485	506F	22	BB 91	SHLD DCSPTR ;STORE THE NEW POINTER VALUE	
486	5072	.	.	;*****	
487	5072	3A	BE 91	LDA DCBPTR+1 ;FETCH UNLOAD POINTER	
488	5075	BC	.	CMP H ;BUFFER OVERFLOW? (IF EQUAL)	
489	5076	C2	80 50	JNZ INT040 ;NO	
490	5079	3A	BD 91	LDA DCBPTR ;CHECK LSBYTES	
491	507C	BD	.	CMP L ;OVERFLOW?	
492	507D	CA	5F 50	JZ INT050 ;YES, STORE ALL ONES	
493	5080	.	.	INT040 EQU \$	
494	5080	.	.	;*****	
495	5080	71	.	MOV M,C ;NO - STORE THE NEW CHARACTE	
496	5081	.	.	;*****	
497	5081	.	.	; EXIT *	
498	5081	.	.	;*****	
499	5081	.	.	INT160 EQU \$	
500	5081	E1	.	POP H ;RESTORE REGISTERS AND	
501	5082	.	.	INT170 EQU \$	
502	5082	C1	.	POP B ;PROCESSOR STATUS	
503	5083	F1	.	POP PSW	
504	5084	FB	.	EI ;RE-ENABLE INTERRUPTS	
505	5085	C9	.	RET ;RETURN	
506	5086	.	.	;	
507	5086	.	.	;*****	
508	5086	.	.	; COMPARE BUFFER START ADDRESS WITH HL. IF EQUAL,	
509	5086	.	.	; SET HL = BUFFER START + BUFFER LENGTH	
510	5086	.	.	; = BUFFER END + 1	
511	5086	.	.	;	
512	5086	.	.	INT180 EQU \$	
513	5086	3A	BA 91	LDA DCBFBG+1 ;COMPARE MSBYTES	
514	5089	BC	.	CMP H	
515	508A	C0	.	RNZ ;NOT EQUAL	
516	508B	3A	B9 91	LDA DCBFBG ;COMPARE LSBYTES	
517	508E	BD	.	CMP L	
518	508F	C0	.	RNZ ;NOT EQUAL	
519	5090	2A	AA 91	LHLD DCBEND ;SET HL = BUFFER END	
520	5093	C9	.	RET	
521	5094	.	.	;*****	


```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 15
=====
523      5094      . . .      ;*****
524      5094      . . .      ; ZCHKTK--CHECK FOR TEK MODE ON
525      5094      . . .      ; EXIT  NZ => TEK MODE ON, A DESTROYED
526      5094      . . .      ;*****
527      5094      . . .      ZCHKTK EQU $
528      5094      3A AD 90      LDA  ZTKFLG
529      5097      E6 41 .      ANI  SCLD+UNSCLD ;EITHER MODE ON?
530      5099      C9 . . .      RET
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 16
=====
532      509A      . . .      ;
533      509A      . . .      ; * * * * *
534      509A      . . .      ;
535      509A      . . .      ; PUTDC - DATACOM OUTPUT ROUTINE
536      509A      . . .      ;
537      509A      . . .      ; ENTRY A = CHAR TO BE OUTPUT
538      509A      . . .      ; NC = NORMAL CHAR
539      509A      . . .      ; C = LAST CHAR IN BLOCK
540      509A      . . .      ;
541      509A      . . .      ; EXIT A DESTROYED
542      509A      . . .      ; NC = NO ERRORS DETECTED
543      509A      . . .      ; Z = CHARACTER ACCEPTED
544      509A      . . .      ; NZ = WAIT
545      509A      . . .      ; C = DATACOM ERROR
546      509A      . . .      ; Z = TRANSMIT MODE, NO ERROR MSG
547      509A      . . .      ;
548      509A      . . .      PUTDC EQU $
549      509A      E5 . .      PUSH H
550      509B      6F . .      MOV L,A ;SAVE CHAR IN L
551      509C      3A AC 91      LDA DCCT ;READ CONTROL WD
552      509F      67 . .      MOV H,A ;SAVE CONTROL IN H
553      50A0      3A B8 91      LDA DCFLGS ;READ FLAGS
554      50A3      0F . .      RRC ;TERMINAL IN RECEIVE?
555      50A4      D2 B2 50      JNC PDC004 ;NO, JMP
556      50A7      21 F7 FF      LXI H,ERRFLG
557      50AA      7E . .      MOV A,M
558      50AB      F6 01 .      ORI DCMERR
559      50AD      77 . .      MOV M,A
560      50AE      E1 . .      POP H
561      50AF      97 . .      SUB A
562      50B0      37 . .      STC
563      50B1      C9 . .      RET ;C,Z => ERROR, NO MESSAGE
564      50B2      . . .      PDC004 EQU $
565      50B2      E6 08 .      ANI FB9600-GOBIN ;FASTBIN ?
566      50B4      3E 2E .      MVI A,B9600+DCNP ;SET FASTBIN MODE
567      50B6      C2 BE 50      JNZ PDC005 ;YES, JUMP
568      50B9      3A FC FF      LDA KBDCSW ;READ SWITCHES
569      50BC      E6 3E .      ANI BAUDPT ;CLEAR H/F AND CA BITS
570      50BE      . . .      PDC005 EQU $
571      50BE      B4 . .      ORA H ;SET COMMAND WORD
572      50BF      32 40 81      STA IODCCT ;OUTPUT COMMAND WORD TO DC
573      50C2      3A 20 81      LDA IODCST ;INPUT DATACOM STATUS
574      50C5      E6 02 .      ANI DCTBE ;IS TRANSMIT BUFFER EMPTY?
575      50C7      C2 D5 50      JNZ PDC020 ;YES, GO OUTPUT THE CHAR
576      50CA      C5 . .      PUSH B
577      50CB      D5 . .      PUSH D
578      50CC      CD 2C 53      CALL DCMON ;AVOID EXTERNAL HANG
579      50CF      D1 . .      POP D
580      50D0      C1 . .      POP B
581      50D1      E1 . .      POP H
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
582	50D2	97	.	.	SUB A	
583	50D3	3C	.	.	INR A	
584	50D4	C9	.	.	RET	;NC,NZ => WAIT
585	50D5	.	.	.	PDC020 EQU \$;OUTPUT CHARACTER
586	50D5	3A	AE	91	LDA FPMASK	;READ FORCE PARITY MASK
587	50D8	B5	.	.	URA L	;OR WITH CHAR
588	50D9	32	60	81	STA IODCDU	;OUTPUT THE CHAR
589	50DC	3A	B8	91	LDA DCFLGS	;NOT BINARY/,
590	50DF	67	.	.	MOV H,A	;SAVE FLAGS
591	50E0	E6	24	.	ANI BINMOD+MCMOD	;AND MC ?
592	50E2	CA	E8	50	JZ PDC030	;YES,
593	50E5	.	.	.	PDC025 EQU \$	
594	50E5	.	.	.	;	NO, RETURN => NC,Z
595	50E5	E1	.	.	POP H	
596	50E6	BF	.	.	CMP A	
597	50E7	C9	.	.	RET	
598	50E8	.	.	.	PDC030 EQU \$	
599	50E8	3A	AD	91	LDA ENDCHR	;IS CHAR ED CHAR?
600	50E8	BD	.	.	CMP L	
601	50EC	CA	49	51	JZ PDC070	;YES, JMP
602	50EF	7C	.	.	MOV A,H	;RECALL DCFLGS
603	50F0	E6	02	.	ANI TRNMOD	;TRANSPARENT?
604	50F2	C2	E5	50	JNZ PDC025	;RETURN ON TRANSPARENT MODE
605	50F5	7D	.	.	MOV A,L	;IS IT LAST CHAR OF BLOCK?
606	50F6	FE	06	.	CPI ACK	;CHAR ACK?
607	50F8	CA	3B	51	JZ PDC060	;YES, RESPOND OUT
608	50FB	3A	F8	FF	LDA KBJMPR	;FETCH KEYBOARD JUMPER A-H
609	50FE	E6	08	.	ANI PAGSTR	;PAGE MODE?
610	5100	C2	11	51	JNZ PDC040	;YES, CHECK FOR DC2 OR RS
611	5103	.	.	.	PDC035 EQU \$	
612	5103	7D	.	.	MOV A,L	;NO, CHECK FOR CR OR LF
613	5104	FE	0D	.	CPI CR	;IS IT CR?
614	5106	CA	28	51	JZ PDC050	;YES, CHK TO SEND END OF DAT
615	5109	FE	0A	.	CPI LF	;IS IT LF?
616	510B	CA	33	51	JZ PDC055	;YES, CHK TO SEND END OF DAT
617	510E	E1	.	.	POP H	
618	510F	BF	.	.	CMP A	;NONE OF THE ABOVE, RETURN
619	5110	C9	.	.	RET	
620	5111	.	.	.	PDC040 EQU \$	
621	5111	7D	.	.	MOV A,L	
622	5112	FE	12	.	CPI DC2	;IS IT DC2?
623	5114	CA	3B	51	JZ PDC060	;YES, SEND END OF DATA
624	5117	3A	F3	FF	LDA MDFLG2	;GET MODE FLAGS
625	511A	E6	02	.	ANI BLKMDE	;ARE WE IN BLOCK MODE?
626	511C	CA	03	51	JZ PDC035	;NO -
627	511F	7D	.	.	MOV A,L	;RESTORE CHAR
628	5120	FE	1E	.	CPI RS	;IS IT RS?
629	5122	CA	3B	51	JZ PDC060	;YES, SEND END OF DATA
630	5125	E1	.	.	POP H	
631	5126	BF	.	.	CMP A	;NONE OF THE ABOVE, RETURN

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE	18
632	5127	C9	.	.	RET		
633	5128	.	.	.	PDC050 EQU \$		
634	5128	3A	F3	FF	LDA MDFLG2		;AUTO LF KEY UP?
635	5128	E6	04	.	ANI AUTOLF		
636	512D	CA	3B	51	JZ PDC060		;YES, SEND END OF DATA
637	5130	E1	.	.	PDC052 POP H		
638	5131	BF	.	.	CMP A		
639	5132	C9	.	.	RET		
640	5133	.	.	.	PDC055 EQU \$		
641	5133	3A	F3	FF	LDA MDFLG2		;AUTO LF KEY DOWN?
642	5136	E6	04	.	ANI AUTOLF		
643	5138	CA	30	51	JZ PDC052		;NO, RETURN
644	513B	.	.	.	PDC060 EQU \$		
645	513B	3A	AD	91	LDA ENDCHR		;GET END OF DATA CHAR
646	513E	CD	9A	50	CALL PUTDC		;AND TRANSMIT
647	5141	DA	3B	51	JC PDC060		
648	5144	C2	3B	51	JNZ PDC060		
649	5147	E1	.	.	POP H		
650	5148	C9	.	.	RET		
651	5149	.	.	.	PDC070 EQU \$		
652	5149	3E	7F	.	MVI A,ADEL		;ALLOW ENOUGH TIME TO MAKE
653	514B	B7	.	.	ORA A		
654	514C	CD	9A	50	CALL PUTDC		;SURE END OF DATA HAS
655	514F	DA	49	51	JC PDC070		;BEEN TRANSMITTED
656	5152	C2	49	51	JNZ PDC070		
657	5155	3E	7F	.	PDC080 MVI A,ADEL		
658	5157	CD	9A	50	CALL PUTDC		
659	515A	DA	55	51	JC PDC080		
660	515D	C2	55	51	JNZ PDC080		
661	5160	C5	.	.	PUSH B		
662	5161	CD	04	54	CALL DCM110		;GO TO RECEIVE STATE
663	5164	C1	.	.	POP B		
664	5165	E1	.	.	POP H		
665	5166	AF	.	.	XRA A		
666	5167	C9	.	.	RET		

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 19
668	5168	. . .	;	
669	5168	. . .	; * * * * *	
670	5168	. . .	;	
671	5168	. . .	; STRTBN - SET BINARY OUTPUT MODE	
672	5168	. . .	;	
673	5168	. . .	; ENTRY DON'T CARE	
674	5168	. . .	;	
675	5168	. . .	; EXIT DON'T CARE	
676	5168	. . .	;	
677	5168	. . .	STRTBN EQU \$	
678	5168	. . .	STB010 EQU \$	
679	5168	3A 20 81	LDA IODCST ;READ DC STATUS	
680	5168	E6 02 .	ANI DCTBE ;XMIT BUFF EMPTY?	
681	516D	CA 68 51	JZ STB010 ;NO, WAIT	
682	5170	3A AC 91	LDA DCCT ;READ CONTROL WD	
683	5173	F6 20 .	ORI DCNP ;SET NO PARITY	
684	5175	32 AC 91	STA DCCT ;SET NEW CONTROL WD	
685	5178	AF . .	XRA A ;SET RETURN INDICATORS	
686	5179	32 AE 91	STA FPMASK ;CLEAR FORCE PARITY	
687	517C	C9 . .	RET	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 20
689	517D	.	.	.	;
690	517D	.	.	.	; INITDC - INITIALIZE DATACOM
691	517D	.	.	.	;
692	517D	.	.	.	INITDC EQU \$
693	517D	.	.	.	;*****
694	517D	.	.	.	; IF EITHER STRAPS P OR Q OUT, ALLOCATE A LARGE
695	517D	.	.	.	; BUFFER (CAUSE OF TEK MODE)
696	517D	21	60	00	LXI H,NRMBUF ;ASSUME NORMAL BUFFER SIZE
697	5180	3A	FA	FF	LDA KBJMP2 ;FETCH KEYBOARD STRAPS
698	5183	E6	60	.	ANI PJMPR+QJMPR ;EITHER STRAP OUT?
699	5185	CA	8B	51	JZ IDC010 ;NO, LEAVE BUFFER SIZE SMALL
700	5188	21	00	08	LXI H,BIGBUF ;YES, ALLOCATE LARGE BUFFER
701	5188	.	.	.	IDC010 EQU \$
702	518B	22	AA	91	SHLD DCBEND ;STORE SIZE TEMPORARILY
703	518E	44	.	.	MOV B,H ;SET BC = BUFFER SIZE
704	518F	4D	.	.	MOV C,L
705	5190	C9	.	.	RET ;GET THE BUFFER
706	5191	.	.	.	;*****
707	5191	.	.	.	;
708	5191	.	.	.	; DATACOM INITIALIZATION CONTINUATOR
709	5191	.	.	.	;
710	5191	.	.	.	INI2DC EQU \$
711	5191	.	.	.	;*****
712	5191	.	.	.	; COMPUTE THE BUFFER END ADDRESS
713	5191	.	.	.	; = BUFFER START + BUFFER LENGTH
714	5191	2A	AA	91	LHLD DCBEND ;HL = LENGTH, DE = START
715	5194	19	.	.	DAD D ;HL = ENDING ADDRESS
716	5195	22	AA	91	SHLD DCBEND ;STORE BUFFER END ADDRESS
717	5198	.	.	.	;*****
718	5198	EB	.	.	XCHG ;PUT BUFFER START IN H AND L
719	5199	22	89	91	SHLD DCBFBG ;STORE BUFFER START ADDRESS
720	519C	.	.	.	RSTDCB EQU \$
721	519C	3A	40	81	LDA IODPCP ;READ GP DC PROG. STRAPS
722	519F	E6	80	.	ANI GPASYC ;GP ASYNC BOARD IN?
723	51A1	3A	F9	FF	LDA KBJMP3 ;GET STRAPS FOR 202 PRCL
724	51A4	67	.	.	MOV H,A ;OPTIONS
725	51A5	C2	AD	51	JNZ RST005 ;YES GP CARD IN
726	51A8	E6	BF	.	ANI 377Q-CHEKCC ;NO, INHIBIT CC LED MONITO
727	51AA	32	F9	FF	STA KBJMP3 ;STORE STRAPS SETTINGS
728	51AD	E6	02	.	RST005 ANI ETXSTP ;USE ETX?
729	51AF	3E	03	.	MVI A,ETX ;USE ETX
730	51B1	C2	B6	51	JNZ RST006 ;YES
731	51B4	3E	04	.	MVI A,EOT ;USE EOT
732	51B6	.	.	.	RST006 EQU \$
733	51B6	32	AD	91	STA ENDCHR ;STORE ED CHAR
734	51B9	7C	.	.	MOV A,H ;RESTORE JMPERS
735	51BA	E6	20	.	ANI SETCH ;IS CH SET?
736	51BC	3E	80	.	MVI A,DCCH ;SET CH OFF
737	51BE	CA	C2	51	JZ RST007 ;NO
738	51C1	AF	.	.	XRA A ;SET CH ON

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  21
=====
 739     51C2      . . .      RST007 EQU $
 740     51C2     32 AC 91      STA DCCT          ;SET CONTROL WORD
 741     51C5     7C . .      MOV A,H          ;RESTORE JUMPERS
 742     51C6      . . .      RSTSRM EQU $
 743     51C6     E6 03 .      ANI MNCHAN      ;MAIN CHANNEL?
 744     51C8     2E 00 .      MVI L,0
 745     51CA     C2 CF 51     JNZ RST010      ;YES,
 746     51CD     2E 20 .      MVI L,MCMOD     ;SET NOT MAIN CHANNEL
 747     51CF     7C . .      RST010 MOV A,H      ;RESTORE JUMPERS
 748     51D0     E6 80 .      ANI FRCPTY      ;SET FORCE PARITY
 749     51D2     32 AE 91     STA FPMASK
 750     51D5     B5 . .      RST020 GRA L     ;SET FP AND MC
 751     51D6     32 B8 91     STA DCFLGS
 752     51D9     3A AC 91     LDA DCCT        ;READ CONTROL WD
 753     51DC     E6 80 .      ANI DCCH        ;CLEAR EXCEPT CH
 754     51DE     32 AC 91     STA DCCT        ;RESTORE
 755     51E1     67 . .      MOV H,A         ;SAVE DCCT
 756     51E2     3A FC FF     LDA KBDCSW      ;READ DC SWITCHES
 757     51E5     E6 3E .      ANI BAUDPT      ;MASK OUT H/F
 758     51E7     B4 . .      GRA H           ;SET CONTROL WORD
 759     51E8     32 40 81     STA IODCCT      ;OUTPUT DC CONTROLS
 760     51EB     3E C3 .      MVI A,JMP       ;SET JUMP OP CODE
 761     51ED     32 B5 91     STA DCMJMP      ;STORE OP
 762     51F0     21 2C 53     LXI H,DCMON     ;SET INITIAL DATA COMM
 763     51F3     22 B6 91     SHLD DCMVEC     ;MONITOR VECTOR
 764     51F6     3A 20 81     LDA IODCST      ;SET INITIAL DATA COMM STATU
 765     51F9     32 BF 91     STA DCSTAT
 766     51FC     3E 01 .      MVI A,1         ;SET THE BLOCK TRANSFER
 767     51FE     CD 7B 54     CALL DCCTL      ;TRIGGER
 768     5201      . . .      RSTDC1 EQU $
 769     5201     2A B9 91     LHL DCBFBG     ;SET LOAD AND UNLOAD POINTER
 770     5204     22 BB 91     SHLD DCSPTR     ;EQUAL TO EACH OTHER
 771     5207     22 BD 91     SHLD DCBPTR
 772     520A     C9 . .      RET
=====

```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 22
774	520B	. . .	;	
775	520B	. . .	; * * * * *	
776	520B	. . .	;	
777	520B	. . .	; GETDC - GET DATA (7 BITS) FROM DATA COM	
778	520B	. . .	;	
779	520B	. . .	; ENTRY DON'T CARE	
780	520B	. . .	;	
781	520B	. . .	; EXIT NC - NO ERRORS DETECTED	
782	520B	. . .	; Z - CHARACTER AVAILABLE	
783	520B	. . .	; A = CHARACTER	
784	520B	. . .	; NZ - WAIT	
785	520B	. . .	; A DESTROYED	
786	520B	. . .	; C - DATACOM ERROR	
787	520B	. . .	; A DESTROYED	
788	520B	. . .	; Z - NO ERROR MESSAGE	
789	520B	. . .	; NZ - DISPLAY ERROR MESSAGE	
790	520B	. . .	; B,C -> ERROR MESSAGE	
791	520B	. . .	;	
792	520B	. . .	GETDC EQU \$	
793	520B	E5 . .	PUSH H	
794	520C	21 B8 91	LXI H,DCFLGS ;FETCH DATACOM FLAGS	
795	520F	F3 . .	DI	
796	5210	7E . .	MOV A,M	
797	5211	E6 FB .	ANI 377Q-BINMOD ;TURN OFF BINARY	
798	5213	77 . .	MOV M,A	
799	5214	E6 80 .	ANI FORPAR ;FORCE PARITY ON?	
800	5216	32 AE 91	STA FPMASK ;SET FPMASK	
801	5219	7E . .	MOV A,M ;RESTORE A	
802	521A	. . .	GDC001 EQU \$	
803	521A	FB . .	EI	
804	521B	E6 42 .	ANI SPECHO+TRNMOD ;IGNORE ECHO FROM DATACO	
805	521D	FE 40 .	CPI SPECHO ;IS SPECHO ON WITHOUT TRANS?	
806	521F	C2 4F 52	JNZ GDC005 ;NO CHECK FOR CHAR	
807	5222	7E . .	MOV A,M ;GET FLAGS AGAIN	
808	5223	E6 05 .	ANI BINMOD+DCCA	
809	5225	FE 01 .	CPI DCCA ;NOT BIN AND REC ?	
810	5227	C2 49 52	JNZ GDC002 ;NO - RETURN WAIT STATUS	
811	522A	3A F9 FF	LDA KBJMP3 ;GET SWITCH SETTINGS	
812	522D	E6 01 .	ANI STXSTP ;TEST FOR SD CHAR	
813	522F	C2 49 52	JNZ GDC002 ;NO CONTINUE WAIT	
814	5232	CD 4E 52	CALL GDC004 ;YES - CHK FOR STRT OF DATA	
815	5235	DA 49 52	JC GDC002 ;NO CHAR, CONTINUE WAIT	
816	5238	C2 49 52	JNZ GDC002 ;NO CHAR, CONTINUE WAIT	
817	5238	FE 02 .	CPI STX ;WAS CHR STX ?	
818	523D	C2 49 52	JNZ GDC002 ;NO, CONTINUE WAITING	
819	5240	21 B8 91	LXI H,DCFLGS ;YES,TURN OFF ECHO SUPPRESS	
820	5243	F3 . .	DI	
821	5244	7E . .	MOV A,M	
822	5245	E6 BF .	ANI 377Q-SPECHO	
823	5247	77 . .	MOV M,A	


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 23
=====
824      5248      FB      .      .      EI
825      5249      .      .      .      ;
826      5249      .      .      .      ; NO DATA - EXIT WAIT
827      5249      .      .      .      ;
828      5249      .      .      .      GDC002 EQU $
829      5249      E1      .      .      POP H ;RESTORE H,L
830      524A      F6      01      .      ORI 1 ;SET NC,NZ
831      524C      FB      .      .      EI
832      524D      C9      .      .      RET ;RETURN WAIT
833      524E      .      .      .      GDC004 EQU $
834      524E      E5      .      .      PUSH H ;SAVE H AND L
835      524F      .      .      .      GDC005 EQU $
836      524F      F3      .      .      DI
837      5250      2A      BD      91      LHLD DCBPTR ;GET THE UNLOAD ADDRESS
838      5253      .      .      .      ;*****
839      5253      3A      BC      91      LDA DCSPTR+1 ;COMPARE WITH LOAD POINTER
840      5256      BC      .      .      CMP H ;ANY CHARACTERS IN BUFFER?
841      5257      C2      70      52      JNZ GDC007 ;YES, GET ONE
842      525A      3A      BB      91      LDA DCSPTR ;ANY CHARS?
843      525D      BD      .      .      CMP L ;(COMPARE LSBYTES)
844      525E      C2      70      52      JNZ GDC007 ;YES, GET ONE
845      5261      .      .      .      ;*****
846      5261      .      .      .      ; NO - CHECK FOR DATA COMM
847      5261      3A      20      81      LDA IODCST ;DATA IN
848      5264      E6      01      .      ANI DCDP ;DATA PRESENT?
849      5266      CA      49      52      JZ GDC002 ;NO - EXIT
850      5269      CD      26      50      CALL DCINTR ;YES - GET THE CHARACTER
851      526C      FB      .      .      EI ;RE-ENABLE INTERRUPTS
852      526D      C3      4F      52      JMP GDC005 ;GET CHAR FROM INPUT BUFFER
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 24
=====
 854      5270      . . .      ;
 855      5270      . . .      ; GET CHARACTER FROM INPUT BUFFER
 856      5270      . . .      ;
 857      5270      . . .      GDC007 EQU $
 858      5270      . . .      ;*****
 859      5270      . . .      ; IF AT BEGINNING OF BUFFER, WRAP AROUND TO END
 860      5270      . . .      ; HL = DCBPTR
 861      5270      CD 86 50      CALL INT180
 862      5273      . . .      ;*****
 863      5273      . . .      ;
 864      5273      . . .      GDC008 EQU $
 865      5273      2B . .      DCX H          ;DECREMENT TO NEXT CHARACTER
 866      5274      22 BD 91      SHLD DCBPTR    ;STORE NEW POINTER
 867      5277      FB . .      EI
 868      5278      7E . .      MOV A,M        ;GET THE INPUT BYTE
 869      5279      6F . .      MOV L,A        ;SAVE IT IN THE L-REGISTER
 870      527A      FE FF . .      CPI 377Q       ;POSSIBLE ERROR BYTE?
 871      527C      C2 8B 52      JNZ GDC010     ;NO - CHECK FOR NORMAL MODE
 872      527F      3A F7 FF      LDA ERRFLG     ;YES - GET ERROR FLAG
 873      5282      E6 01 . .      ANI DCMERR     ;ERROR IN INPUT?
 874      5284      CA 8B 52      JZ GDC010      ;NO - PROCESS CHARACTER
 875      5287      E1 . .      POP H         ;YES - RESTORE H,L
 876      5288      BF . .      CMP A         ;RETURN ERROR WITH NO MESSAG
 877      5289      37 . .      STC          ;(C, Z)
 878      528A      C9 . .      RET          ;RETURN
 879      528B      . . .      ;
 880      528B      . . .      ; PROCESS CHARACTER
 881      528B      . . .      ;
 882      528B      . . .      ;
 883      528B      3A B8 91      GDC010 EQU $
 884      528E      67 . .      LDA DCFLGS    ;GET DATACOM FLAGS
 885      528F      E6 04 . .      MOV H,A       ;SAVE FLAGS
 886      5291      C2 11 53      ANI BINMOD    ;BINARY MODE?
 887      5294      7D . .      JNZ GDC050    ;YES - DON'T CHECK FOR <ENQ>
 888      5295      E6 7F . .      MOV A,L       ;MASK OUT PARITY BIT
 889      5297      6F . .      MOV L,A       ;SAVE CURRENT CHAR
 890      5298      7C . .      MOV A,H       ;RECALL FLAGS
 891      5299      E6 20 . .      ANI MCMOD     ;MAIN CHANNEL?
 892      529B      C2 D3 52      JNZ GDC015    ;NO, DO NOT CK ED CHAR
 893      529E      3A AD 91      LDA ENDCHR    ;GET ED CHAR
 894      52A1      BD . .      CMP L
 895      52A2      C2 D3 52      JNZ GDC015    ;NO, CONTINUE
 896      52A5      7C . .      MOV A,H       ;RECALL STRAPS
 897      52A6      E6 02 . .      ANI TRNMOD    ;TRANSPARENT?
 898      52A8      C2 C8 52      JNZ GDC014    ;YES JMP
 899      52AB      3A F9 FF      LDA KBJMP3    ;READ STRAPS
 900      52AE      E6 08 . .      ANI CFSTRP    ;CF DETECT ENABLED ?
 901      52B0      C2 BF 52      JNZ GDC012    ;NO, GO TO XMIT STATE
 902      52B3      21 B8 91      LXI H,DCFLGS ;GET DCFLGS ADDR
 903      52B6      F3 . .      DI
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
						25
904	52B7	7E	.	.	MOV A,M ;GET FLAGS	
905	52B8	F6	40	.	ORI SPECHO ;SET SUPPRESS ECHO	
906	52BA	77	.	.	MOV M,A ;STORE NEW FLAGS	
907	52BB	FB	.	.	EI	
908	52BC	C3	49	52	JMP GDC002 ;GO EXIT ON WAIT DROP CF	
909	52BF	C5	.	.	GDC012 PUSH B ;SAVE B,C	
910	52C0	CD	75	53	CALL DCM030 ;YES, GO TO TRANSMIT STATE	
911	52C3	C1	.	.	POP B	
912	52C4	E1	.	.	POP H	
913	52C5	F6	01	.	ORI 1	
914	52C7	C9	.	.	RET	
915	52C8	.	.	.	;	
916	52C8	.	.	.	GDC014 EQU \$	
917	52C8	C5	.	.	PUSH B	
918	52C9	CD	75	53	CALL DCM030	
919	52CC	C1	.	.	POP B	
920	52CD	E1	.	.	POP H	
921	52CE	3A	AD	91	LDA ENDCHR ;GET ED CHAR	
922	52D1	BF	.	.	CMP A	
923	52D2	C9	.	.	RET	
924	52D3	.	.	.	;	
925	52D3	.	.	.	GDC015 EQU \$	
926	52D3	7C	.	.	MOV A,H ;RECALL FLAGS	
927	52D4	E6	42	.	ANI TRNMOD+SPECHO ;TRANSPARENT OR WAITING	
928	52D6	C2	11	53	JNZ GDC050 ;YES, JMP	
929	52D9	7D	.	.	MOV A,L ;RECALL CHAR	
930	52DA	FE	05	.	CPI ENQ ;IS IT AN <ENQ>?	
931	52DC	.	.	.	*****	
932	52DC	.	.	.	; IF IN TEK MODE, DONT RESPOND WITH ACK TO ENQ	
933	52DC	CC	94	50	CZ ZCHKTK ;(TEST FOR TEK MODE)	
934	52DF	C2	11	53	JNZ GDC050 ;NOT ENQ OR ENQ BUT TEK MODE	
935	52E2	E1	.	.	POP H ;IS ENQ, RESTORE H	
936	52E3	.	.	.	*****	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 26
=====
 938      52E3      . . . ;
 939      52E3      . . . ; WAIT FOR LINE TURNAROUND BEFORE SENDING <ACK>
 940      52E3      . . . ;
 941      52E3      . . . GDC020 EQU $
 942      52E3      CD 4E 52      CALL GDC004      ;CHECK FOR END OF DATA
 943      52E6      3A 88 91      LDA DCFLGS      ;GET THE DATACOM FLAGS
 944      52E9      E6 01 .       ANI DCCA        ;TRANSMIT MODE (CA = 1)?
 945      52EB      C2 E3 52      JNZ GDC020      ;NO - WAIT FOR TURNAROUND
 946      52EE      . . . GDC030 EQU $      ;YES - SEND ACK
 947      52EE      3E 06 .       MVI A,ACK
 948      52F0      B7 . .       ORA A           ;NC => NORMAL CHAR
 949      52F1      CD 9A 50      CALL PUTDC
 950      52F4      DA EE 52      JC GDC030      ;ERROR - TRY AGAIN
 951      52F7      C2 EE 52      JNZ GDC030      ;WAIT - TRY AGAIN
 952      52FA      3A 88 91      LDA DCFLGS      ;SEE IF ENQ-ACK SHOULD CAUSE
 953      52FD      2F . .       CMA             ;CHANGE TO BINARY MODE
 954      52FE      E6 08 .       ANI GOBIN
 955      5300      C0 . .       RNZ             ;NO - RET WITH NC,NZ => WAIT
 956      5301      F3 . .       DI
 957      5302      3A 88 91      LDA DCFLGS      ;YES - TOGGLE GOBIN AND BINM
 958      5305      EE 0C .       XRI GOBIN+BINMOD
 959      5307      32 B8 91      STA DCFLGS
 960      530A      FB . .       EI
 961      530B      AF . .       XRA A           ;CLEAR FP MASK
 962      530C      32 AE 91      STA FPMASK
 963      530F      3C . .       INR A
 964      5310      C9 . .       RET
 965      5311      . . . ;
 966      5311      . . . ; BINARY OR TRANSPARENT - RETURN CHARACTER
 967      5311      . . . ;
 968      5311      . . . GDC050 EQU $
 969      5311      7D . .       MOV A,L         ;(PUT CHARACTER IN A-REG)
 970      5312      E1 . .       POP H
 971      5313      BF . .       CMP A           ;RETURN CHARACTER
 972      5314      C9 . .       RET             ;(NC, Z)
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	
974	5315	.	.	.	;
975	5315	.	.	.	; * * * * *
976	5315	.	.	.	;
977	5315	.	.	.	; GETBIN - GET A BINARY BYTE
978	5315	.	.	.	;
979	5315	.	.	.	; ENTRY DON'T CARE
980	5315	.	.	.	;
981	5315	.	.	.	; EXIT SAME AS GETDC, ABOVE
982	5315	.	.	.	;
983	5315	.	.	.	GETBIN EQU \$
984	5315	E5	.	.	PUSH H
985	5316	21	B8	91	LXI H,DCFLGS
986	5319	7E	.	.	MOV A,M ;LOAD DC FLAGS
987	531A	E6	04	.	ANI BINMOD ;ALREADY IN BINARY MODE?
988	531C	F3	.	.	UI
989	531D	7E	.	.	MOV A,M
990	531E	C2	1A	52	JNZ GDC001 ;YES - GET BYTE AND RETURN
991	5321	F6	08	.	ORI GOBIN ;NO - SET FLAG TO GO TO BIN
992	5323	77	.	.	MOV M,A
993	5324	F8	.	.	EI
994	5325	E1	.	.	POP H
995	5326	CD	0B	52	CALL GETDC ;EMPTY BUF/DO ENQ-ACK HANDSH
996	5329	F6	01	.	ORI 1 ;NC,NZ => WAIT
997	532B	C9	.	.	RET

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
999	532C	.	.	.	28
1000	532C	.	.	.	
1001	532C	.	.	.	
1002	532C	.	.	.	
1003	532C	.	.	.	
1004	532C	.	.	.	
1005	532C	.	.	.	
1006	532C	.	.	.	
1007	532C	.	.	.	
1008	532C	.	.	.	
1009	532C	.	.	.	
1010	532C	.	.	.	
1011	532C	.	.	.	
1012	532C	.	.	.	
1013	532C	.	.	.	
1014	532C	.	.	.	
1015	532C	.	.	.	
1016	532C	.	.	.	
1017	532C	.	.	.	
1018	532C	.	.	.	
1019	532C	.	.	.	
1020	532C	.	.	.	
1021	532C	.	.	.	
1022	532C	.	.	.	
1023	532C	.	.	.	
1024	532C	.	.	.	
1025	532C	.	.	.	
1026	532C	.	.	.	
1027	532C	.	.	.	
1028	532C	.	.	.	
1029	532C	.	.	.	
1030	532C	.	.	.	
1031	532C	.	.	.	
1032	532C	.	.	.	
1033	532C	.	.	.	
1034	532C	.	.	.	
1035	532C	.	.	.	
1036	532C	.	.	.	
1037	532C	.	.	.	
1038	532C	.	.	.	
1039	532C	.	.	.	
1040	532C	.	.	.	
1041	532C	.	.	.	
1042	532C	3A	AC	91	
1043	532F	57	.	.	
1044	5330	3A	B8	91	
1045	5333	E6	10	.	
1046	5335	3E	2E	.	
1047	5337	C2	3F	53	
1048	533A	3A	FC	FF	

```

DCMON EQU $
LDA DCCT ;READ CONTROL WD
MOV D,A ;SAVE CH,NP,CA
LDA DCFLGS ;GET DATA COMM FLAGS
ANI FB9600 ;FAST BINARY?
MVI A,B9600+DCNP ;SET FASTBIN
JNZ DCM002 ;YES,
LDA KBDCSW ;READ DCSW
    
```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	
1049	533D	E6	3E	.	ANI BAUDPT	;CLEAR H/F AND CA BITS
1050	533F	B2	.	.	DCM002 ORA D	;KEYBOARD DATACOM SWITCHES
1051	5340	32	40	81	STA IODCCT	;SET BAUD RATE AND PARITY
1052	5343	3A	20	81	LDA IODCST	;GET CURRENT DATA COMM STATU
1053	5346	4F	.	.	MOV C,A	;SAVE IT IN C
1054	5347	21	BF	91	LXI H,DCSTAT	
1055	534A	46	.	.	MOV B,M	;GET PREVIOUS STATUS AND
1056	534B	77	.	.	MOV M,A	;SAVE NEW STATUS
1057	534C	3A	F9	FF	LDA KBJMP3	;READ STRAPS
1058	534F	5F	.	.	MOV E,A	;SAVE IN E
1059	5350	E6	40	.	ANI CHEKCC	;MONITOR CC ?
1060	5352	CA	5D	53	JZ DCM005	;NO USE CB
1061	5355	3A	21	81	LDA IODCS2	;READ 2ND STATUS
1062	5358	E6	80	.	ANI DCCC	;IS CC HIGH ?
1063	535A	C3	60	53	JMP DCM010	;GO TO SET LED
1064	535D	79	.	.	DCM005 MOV A,C	;RECALL STATUS
1065	535E	E6	20	.	ANI DCCB	;IS CB HIGH (DCCB = 0)?
1066	5360	CC	17	48	DCM010 CZ ZSTXMT	;YES - TURN ON TRANSMIT LED
1067	5363	C4	1A	48	CNZ ZCLXMT	;NO - TURN OFF TRANSMIT LED
1068	5366	7A	.	.	MOV A,D	;IN 202 RECEIVE MODE (CA = 0
1069	5367	0F	.	.	RRC	;DCCA = 1)?
1070	5368	7B	.	.	MOV A,E	;RECALL STRAPS
1071	5369	D2	E8	53	JNC DCM100	;NO - CHECK FOR DROP IN SB

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 30
1073	536C	.	.	;	
1074	536C	.	.	;	IN RECEIVE MODE (DCCA = 1 => CA = 0) - CHECK
1075	536C	.	.	;	FOR DROP IN CF TO GO TO TRANSMIT MODE
1076	536C	.	.	;	IF CARRIER DETECT IS ENABLED
1077	536C	.	.	;	
1078	536C	E6	08	.	ANI CFSTRP ;CARRIER DETECT ENABLED?
1079	536E	C0	.	.	RNZ ;NO , IGNORE CF
1080	536F	78	.	.	MOV A,B ;PUT CURRENT STATUS IN A
1081	5370	A9	.	.	XRA C ;EXTRACT CHANGED BITS
1082	5371	E6	10	.	ANI DCCF ;DID RECEIVE CARRIER (CF)
1083	5373	A1	.	.	ANA C ;DROP (DCCF -> 1)?
1084	5374	C8	.	.	RZ ;NO - RETURN
1085	5375	.	.	.	DCM030 EQU \$
1086	5375	21	B8	91	LXI H,DCFLGS ;YES - PREPARE TO GO INTO
1087	5378	F3	.	.	DI
1088	5379	7E	.	.	MOV A,M ;TRANSMIT MODE
1089	537A	F6	40	.	ORI SPECHO ;SET FLAG TO IGNORE INPUT
1090	537C	77	.	.	MOV M,A ;FROM DATA COMM
1091	537D	FB	.	.	EI
1092	537E	AF	.	.	XRA A ;CLEAR TRANSMIT TURN AROUND
1093	537F	32	B4	91	STA XMTDLY ;TIME LIMIT
1094	5382	3E	01	.	MVI A,1
1095	5384	.	.	.	DCM040 EQU \$
1096	5384	06	00	.	MVI B,0 ;SET TO TURN ON CA
1097	5386	21	8C	53	LXI H,DCM050 ;SET CLEAN-UP ROUTINE
1098	5389	C3	52	54	JMP DCTURN ;DO TURN AROUND
1099	538C	.	.	.	;
1100	538C	.	.	.	;
1101	538C	.	.	.	;
1102	538C	.	.	.	DCM050 EQU \$
1103	538C	3A	20	81	LDA IODCST ;GET CURRENT STATUS
1104	538F	32	BF	91	STA DCSTAT ;UPDATE STATUS
1105	5392	47	.	.	MOV B,A
1106	5393	E6	20	.	ANI DCCB ;IS CB ON? (=0)
1107	5395	C2	A6	53	JNZ DCM055 ;NO, CONTINUE WAIT
1108	5398	3A	FA	FF	LDA KBJMP2 ;MONITOR SB?
1109	5398	E6	80	.	ANI SBSTRP
1110	539D	C2	AE	53	JNZ DCM060 ;NO, GO TO TRANSMIT STATE
1111	53A0	78	.	.	MOV A,B ;YES, IS SB ON? (=0)
1112	53A1	E6	40	.	ANI DCSB
1113	53A3	CA	AE	53	JZ DCM060 ;YES,GO TO TRANSMIT STATE
1114	53A6	.	.	.	DCM055 EQU \$
1115	53A6	21	B4	91	LXI H,XMTDLY ;TURN AROUND TIME LIMIT
1116	53A9	35	.	.	DCR M ;EXCEEDED?
1117	53AA	CA	04	54	JZ DCM110 ;YES - REVERT TO RECEIVE MOD
1118	53AD	C9	.	.	RET ;NO - RETURN (CONTINUE WAIT)
1119	53AE	.	.	.	;
1120	53AE	.	.	.	;
1121	53AE	.	.	.	;
1122	53AE	.	.	.	DCM060 EQU \$;SET TRANSMIT MODE CA

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE
1123	53AE	21	B8	91	LXI H,DCFLGS	31
1124	53B1	F3	.	.	DI	
1125	53B2	7E	.	.	MOV A,M	
1126	53B3	E6	FE	.	ANI 377Q-DCCA	
1127	53B5	77	.	.	MOV M,A	
1128	53B6	21	AC	91	LXI H,DCCT ;SET CONTROL WD ADDR	
1129	53B9	7E	.	.	MOV A,M ;READ WD	
1130	53BA	E6	A0	.	ANI DCCH+DCNP ;SET CA LOW	
1131	53BC	77	.	.	MOV M,A ;SET NEW WD	
1132	53BD	21	2C	53	LXI H,DCMON ;SET MONITOR VECTOR TO	
1133	53C0	22	B6	91	SHLD DCMVEC ;REGULAR MONITOR ROUTINE	
1134	53C3	21	F8	FF	LXI H,CMFLGS	
1135	53C6	7E	.	.	MOV A,M	
1136	53C7	F6	01	.	ORI BLKTRG ;SET BLOCK TRANSFER TRIGGER	
1137	53C9	77	.	.	MOV M,A	
1138	53CA	FB	.	.	EI	
1139	53CB	3A	F9	FF	LDA KBJMP3 ;MAIN CHANNEL PROTOCOL?	
1140	53CE	47	.	.	MOV B,A	
1141	53CF	E6	03	.	ANI MNCHAN	
1142	53D1	C8	.	.	RZ ;NO, RETURN	
1143	53D2	78	.	.	MOV A,B ;YES, SEND START OF DATA?	
1144	53D3	E6	01	.	ANI STXSTP	
1145	53D5	C0	.	.	RNZ ;NO, RETURN	
1146	53D6	.	.	.	DCM070 EQU \$	
1147	53D6	3A	B8	91	LDA DCFLGS ;READ FLAGS	
1148	53D9	E6	02	.	ANI TRNMOD ;TRANSPARENT?	
1149	53DB	C0	.	.	RNZ ;YES - RETURN	
1150	53DC	.	.	.	DCM075 EQU \$	
1151	53DC	3E	02	.	MVI A,STX ;SET STX AS SD	
1152	53DE	CD	9A	50	CALL PUTDC	
1153	53E1	DA	DC	53	JC DCM075	
1154	53E4	C2	DC	53	JNZ DCM075	
1155	53E7	C9	.	.	RET ;RETURN	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 32
1157	53E8	.	.	;	
1158	53E8	.	.	; IN TRANSMIT MODE (DCCA = 0 => CA = 1) - CHECK	
1159	53E8	.	.	; FOR DROP IN SB TO GO TO RECEIVE MODE	
1160	53E8	.	.	;	
1161	53E8	.	.	DCM100 EQU \$	
1162	53E8	E6	04	ANI CBKSTP	
1163	53EA	C0	.	RNZ	
1164	53EB	78	.	MOV A,B	
1165	53EC	A9	.	XRA C	;EXTRACT CHANGED BITS
1166	53ED	E6	40	ANI DCSB	;DID SECONDARY RECEIVED DATA
1167	53EF	A1	.	ANA C	;(SB) DROP (DCSB -> 1)?
1168	53F0	C8	.	RZ	;NO - RETURN
1169	53F1	7B	.	MOV A,E	
1170	53F2	E6	03	ANI MNCHAN	;MAIN CHANNEL?
1171	53F4	CA	04 54	JZ DCM110	;NO,
1172	53F7	.	.	DCM105 EQU \$	
1173	53F7	3A	AD 91	LDA ENDCHR	;OUTPUT ED CHAR
1174	53FA	CD	9A 50	CALL PUTDC	
1175	53FD	DA	F7 53	JC DCM105	
1176	5400	C2	F7 53	JNZ DCM105	
1177	5403	C9	.	RET	
1178	5404	.	.	;	
1179	5404	.	.	DCM110 EQU \$;YES - GO TO RECEIVE MODE
1180	5404	21	F8 FF	LXI H,DCMFLGS	
1181	5407	F3	.	DI	
1182	5408	7E	.	MOV A,M	;CLEAR BLOCK TRANSFER TRIGGE
1183	5409	E6	FE	ANI 377Q-BLKTRG	
1184	5408	77	.	MOV M,A	
1185	540C	06	01	MVI B,DCCA	;SET TO TURN OFF CA
1186	540E	21	B8 91	LXI H,DCFLGS	;SET CA IN "DCFLGS" TO
1187	5411	7E	.	MOV A,M	;INHIBIT TRANSMISSION
1188	5412	B0	.	ORA B	;OF DATA
1189	5413	77	.	MOV M,A	
1190	5414	21	AC 91	LXI H,DCCT	;SET CONTROL WD ADDR
1191	5417	7E	.	MOV A,M	
1192	5418	F6	01	ORI DCCA	;SET CA
1193	541A	77	.	MOV M,A	
1194	541B	FB	.	EI	
1195	541C	3E	01	MVI A,1	;SET SETTLING TIME (10 MSEC)
1196	541E	21	24 54	LXI H,DCM150	;SET CLEAN-UP ROUTINE
1197	5421	C3	52 54	JMP DCTURN	;DO TURN AROUND
1198	5424	.	.	;	
1199	5424	.	.	; TURN AROUND DONE - SET FOR RECEIVE MODE	
1200	5424	.	.	;	
1201	5424	.	.	DCM150 EQU \$	
1202	5424	21	2C 53	LXI H,DCMON	;SET MONITOR VECTOR TO
1203	5427	22	86 91	SHLD DCMVEC	;REGULAR MONITOR ROUTINE
1204	542A	F3	.	DI	
1205	542B	2A	B9 91	LHLD DCBFBG	;SET DATA COMM BUFFER
1206	542E	22	BB 91	SHLD DCSPTR	;POINTERS

13255

2648A MICROCODE LISTING 'DC16'

13255/90010

REV 04/17/78

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 33
=====
1207      5431      22  BD  91      SHLD DCBPTR
1208      5434      FB  .  .      EI
1209      5435      3A  20  81      LDA  IODCST      ;READ STATUS
1210      5438      32  BF  91      STA  DCSTAT      ;UPDATE STATUS
1211      5438      3A  F9  FF      LDA  KBJMP3      ;MAIN CHANNEL PROTOCOL?
1212      543E      47  .  .      MOV  B,A
1213      543F      E6  03  .      ANI  MNCHAN
1214      5441      78  .  .      MOV  A,B
1215      5442      CA  48  54      JZ   DCM160      ;NO, RESET ECHO SUPPRESS
1216      5445      E6  01  .      ANI  STXSTP      ;YES, WAIT FOR STRT OF DATA?
1217      5447      C8  .  .      RZ
1218      5448      .  .  .      DCM160 EQU  $
1219      5448      21  B8  91      LXI  H,DCFLGS   ;SET DATA COMM FLAGS TO
1220      5448      F3  .  .      DI
1221      544C      7E  .  .      MOV  A,M        ;ACCEPT DATA COMM INPUT
1222      544D      E6  BF  .      ANI  377Q-SPECHO
1223      544F      77  .  .      MOV  M,A
1224      5450      FB  .  .      EI
1225      5451      C9  .  .      RET             ;RETURN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 34
1227	5452	.	.	.	;
1228	5452	.	.	.	; * * * * *
1229	5452	.	.	.	;
1230	5452	.	.	.	; DCTURN - TURN LINE AROUND FOR 202
1231	5452	.	.	.	;
1232	5452	.	.	.	; ENTRY A = SIGNAL SETTling TIME
1233	5452	.	.	.	; B = SETTING FOR CA
1234	5452	.	.	.	; H,L = TURN AROUND EXIT ROUTINE
1235	5452	.	.	.	;
1236	5452	.	.	.	; EXIT A,B,H,L DESTROYED
1237	5452	.	.	.	;
1238	5452	.	.	.	DCTURN EQU \$
1239	5452	32	B3	91	STA DCDELAY ;SET SIGNAL SETTling DELAY
1240	5455	22	B1	91	SHLD DCTEX ;SET DATA COMM EXIT ROUTINE
1241	5458	21	6F	54	LXI H,DCDCNT ;SET MONITOR JUMP FOR DELAY
1242	545B	22	B6	91	SHLD DCMVEC ;CONTINUATOR
1243	545E	3A	AC	91	LDA DCCT ;READ CONTROL WD
1244	5461	E6	FE	.	ANI 377Q-DCCA ;SET CLEAR
1245	5463	B0	.	.	ORA B ;MASK WITH DCCA
1246	5464	47	.	.	MOV B,A ;SAVE IN B
1247	5465	.	.	.	DCT010 EQU \$
1248	5465	3A	FC	FF	LDA KBDCSW ;ADD DESIRED CA SETTING WITH
1249	5468	E6	3E	.	ANI BAUDPT ;CLEAR H/F AND CA BITS
1250	546A	B0	.	.	ORA B ;KEYBOARD DATACOM SWITCHES
1251	546B	32	40	81	STA IODCCT ;SET DATA COMM INTERFACE
1252	546E	C9	.	.	RET ;RETURN
1253	546F	.	.	.	;
1254	546F	.	.	.	; TURN AROUND CONTINUATOR - CHECK FOR TIME OUT
1255	546F	.	.	.	;
1256	546F	.	.	.	DCDCNT EQU \$
1257	546F	21	B3	91	LXI H,DCDELAY
1258	5472	35	.	.	DCR M ;SETTLING TIME COMPLETED?
1259	5473	C0	.	.	RNZ ;NO - RETURN
1260	5474	2A	B1	91	LHLD DCTEX ;YES - GET CLEAN UP ROUTINE
1261	5477	22	B6	91	SHLD DCMVEC ;ADDRESS AND GO DO IT
1262	547A	E9	.	.	PCHL

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 35
1264	547B	. . .	;	
1265	547B	. . .	; * * * * *	
1266	547B	. . .	;	
1267	547B	. . .	; DCCTL - PERFORM CONTROL FUNCTIONS	
1268	547B	. . .	;	
1269	547B	. . .	; ENTRY A = CONTROL PARAMETER	
1270	547B	. . .	; B,C = CONTROL VARIABLES (AS NEEDED)	
1271	547B	. . .	;	
1272	547B	. . .	; EXIT A DESTROYED	
1273	547B	. . .	; NC - NO DATACOM ERRORS DETECTED	
1274	547B	. . .	; Z - CONTROL PERFORMED	
1275	547B	. . .	; NZ - INVALID CONTROL REQUEST	
1276	547B	. . .	; C - DATACOM ERROR DETECTED	
1277	547B	. . .	; Z - NO ERROR MESSAGE	
1278	547B	. . .	; NZ - DISPLAY ERROR MESSAGE	
1279	547B	. . .	; B,C = POINTER TO ERROR MESSAGE	
1280	547B	. . .	;	
1281	547B	. . .	DCCTL EQU \$	
1282	547B	FE 0D .	CPI PROMPT ;IS CONTROL CALL PROMPT?	
1283	547D	CA D4 55	JZ SNDDC2 ;YES, JMP	
1284	5480	D0 . .	RNC ;IF GREATER RET INVALID CNTL	
1285	5481	E5 . .	PUSH H ;SAVE THE WORKING REGISTERS	
1286	5482	D5 . .	PUSH D	
1287	5483	87 . .	ADD A ;DOUBLE THE PARAMETER VALUE	
1288	5484	5F . .	MOV E,A ;COMPUTE CONTROL VECTOR	
1289	5485	16 00 .	MVI D,0 ;LOCATION	
1290	5487	21 91 54	LXI H,DCCTAB	
1291	548A	19 . .	DAD D	
1292	5488	5E . .	MOV E,M ;FETCH THE CONTROL VECTOR	
1293	548C	23 . .	INX H	
1294	548D	56 . .	MOV D,M	
1295	548E	EB . .	XCHG	
1296	548F	D1 . .	POP D ;RECALL D AND E	
1297	5490	E9 . .	PCHL ;GO TO CONTROL ROUTINE	
1298	5491	. . .	;	
1299	5491	. . .	; CONTROL VECTORS	
1300	5491	. . .	;	
1301	5491	. . .	DCCTAB EQU \$	
1302	5491	AC 54 .	DW CLBLTR ;0 - CLEAR BLOCK XFR TRIGGER	
1303	5493	B6 54 .	DW STBLTR ;1 - SET BLOCK XFR TRIGGER	
1304	5495	C8 54 .	DW RSTDCM ;2 - RESET DATACOM	
1305	5497	CE 54 .	DW STRMTE ;3 - SET REMOTE MODE	
1306	5499	B3 54 .	DW STLOCL ;4 - SET LOCAL MODE	
1307	549B	DA 54 .	DW BREAK ;5 - OUTPUT BREAK SIGNAL	
1308	549D	1B 55 .	DW DSCNCT ;6 - MODEM DISCONNECT	
1309	549F	00 56 .	DW SNDTRM ;7 - SEND ED IF MAIN CHAN	
1310	54A1	88 55 .	DW ENTMON ;8 - ENTER MONITOR MODE	
1311	54A3	98 55 .	DW RETNRM ;9 - RETURN TO NORMAL OPER	
1312	54A5	A9 55 .	DW FSTBIN ;10 - FAST BINARY 9600 BAUD	
1313	54A7	49 52 .	DW GDC002 ;11 - NO OP	

13255

2648A MICROCODE LISTING 'DC16'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC   OBJECT CODE  SOURCE STATEMENTS                                PAGE 36
=====
1314     54A9   49 52 .           DW  GDC002   ;12 - NO OP
1315     54AB   . . . ;           DFAD SNDDC2  13 - SEND DC2
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1317	54AB	.	.	;	37
1318	54AB	.	.	; CLBLTR - CLEAR BLOCK TRANSFER TRIGGER	
1319	54AB	.	.	;	
1320	54AB	.	.	CLBKTX EQU \$	
1321	54AB	E5	.	PUSH H ;ENTRY FOR SEND DC2 ROUTINE	
1322	54AC	.	.	CLBLTR EQU \$	
1323	54AC	3E	FE	MVI A,377Q-BLKTRG ;SET FLAG TO BE CLEARED	
1324	54AE	21	F8	FF LXI H,CMFLGS ;SET H,L TO COMMON FLAGS	
1325	54B1	A6	.	ANA M ;MASK OUT THE FLAG	
1326	54B2	77	.	MOV M,A ;STORE UPDATED VALUE	
1327	54B3	.	.	DCCTX1 EQU \$	
1328	54B3	E1	.	POP H ;RESTORE H AND L	
1329	54B4	BF	.	CMP \A ;SET NC AND Z	
1330	54B5	C9	.	RET ;RETURN	
1331	54B6	.	.	;	
1332	54B6	.	.	;	
1333	54B6	.	.	;	
1334	54B6	.	.	; STBLTR - SET BLOCK TRANSFER TRIGGER	
1335	54B6	.	.	;	
1336	54B6	.	.	STBLTR EQU \$	
1337	54B6	3A	B8	91 LDA DCFLGS ;GET THE DATACOM FLAGS	
1338	54B9	E6	01	ANI DCCA ;IN 202 RECEIVE MODE?	
1339	54BB	C2	B3	54 JNZ DCCTX1 ;YES - RETURN WO/SETTING FLA	
1340	54BE	3E	01	MVI A,BLKTRG ;NO - SET THE BLOCK TRANSFER	
1341	54C0	21	F8	FF LXI H,CMFLGS ;TRIGGER	
1342	54C3	B6	.	ORA M	
1343	54C4	77	.	MOV M,A	
1344	54C5	C3	B3	54 JMP DCCTX1 ;RETURN	
1345	54C8	.	.	;	
1346	54C8	.	.	; RSTDCM - RESET DATACOM	
1347	54C8	.	.	;	
1348	54C8	.	.	RSTDCM EQU \$	
1349	54C8	CD	9C	51 CALL RSTDCB ;RESET DATACOM FLAGS AND PTR	
1350	54CB	C3	B3	54 JMP DCCTX1 ;RETURN	
1351	54CE	.	.	;	
1352	54CE	.	.	; STRMTE - SET REMOTE MODE	
1353	54CE	.	.	;	
1354	54CE	.	.	STRMTE EQU \$	
1355	54CE	F3	.	DI	
1356	54CF	3A	F9	FF LDA KBJMP3 ;READ DC JUMPERS	
1357	54D2	67	.	MOV H,A ;SAVE IN H	
1358	54D3	CD	C6	51 CALL RSTSRM ;DO PARTIAL RESET	
1359	54D6	FB	.	EI	
1360	54D7	C3	B3	54 JMP DCCTX1 ;RETURN	
1361	54DA	.	.	;	
1362	54DA	.	.	; STLOCL - SET LOCAL MODE	
1363	54DA	.	.	;	
1364	54B3	.	.	STLOCL EQU DCCTX1 ;NO FUNCTION	

13255
2648A MICROCODE LISTING 'DC16'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 39
=====
1416     5526     C3 F1 54          JMP BRK010      ;SET MONITOR ROUTINE JUMP
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE	STATEMENTS	PAGE	40
1418	5529	.	.	.	;		
1419	5529	.	.	.	; TRMBIN - TERMINATE BINARY OUTPUT		
1420	5529	.	.	.	;		
1421	5529	.	.	.	TRMBIN EQU \$		
1422	5529	E5	.	.	PUSH H		
1423	552A	21	B8	91	LXI H,DCFLGS ;CLEAR FAST BINARY		
1424	552D	F3	.	.	DI		
1425	552E	7E	.	.	MOV A,M		
1426	552F	E6	EF	.	ANI 377Q-FB9600		
1427	5531	77	.	.	MOV M,A		
1428	5532	FB	.	.	EI		
1429	5533	E6	80	.	ANI FORPAR ;SET FP MASK		
1430	5535	32	AE	91	STA FPMASK		
1431	5538	3A	40	81	TRM005 LDA IODCPC ;GET DATA COMM PROGRAM STRAP		
1432	553B	E6	80	.	ANI GPASYC ;IS IT THE GP ASYNC BOARD?		
1433	553D	CA	53	55	JZ TRM015 ;NO		
1434	5540	.	.	.	TRM010 EQU \$;YES - DROP REQUEST TO SEND		
1435	5540	3A	21	81	LDA IODCS2 ;GET AUXILARY STATUS		
1436	5543	E6	02	.	ANI DCTBE ;TRANSMIT BUFFER EMPTY?		
1437	5545	CA	40	55	JZ TRM010 ;NO - CONTINUE WAITING		
1438	5548	CD	C0	55	CALL CHKDCM ;MAKE SURE DATA COMM IS IDLE		
1439	5548	3E	01	.	MVI A,1 ;DELAY FOR 10 MSEC		
1440	554D	.	.	.	TRM012 EQU \$		
1441	554D	21	63	55	LXI H,TRM020 ;SET EXIT		
1442	5550	C3	F1	54	JMP BRK010 ;SET MONITOR JUMP ADDRESS		
1443	5553	.	.	.	;		
1444	5553	.	.	.	TRM015 EQU \$		
1445	5553	3A	20	81	LDA IODCST ;READ STATUS		
1446	5556	E6	02	.	ANI DCTBE ;IS XMIT BUFFER EMPTY?		
1447	5558	CA	53	55	JZ TRM015 ;NO		
1448	5558	CD	C0	55	CALL CHKDCM ;CHECK DATA COMM IDLE		
1449	555E	3E	0A	.	MVI A,10 ;SET DELAY 100 MSEC		
1450	5560	C3	4D	55	JMP TRM012		
1451	5563	.	.	.	;		
1452	5563	.	.	.	; CONTINUATION FOR BINARY TERMINATE		
1453	5563	.	.	.	;		
1454	5563	.	.	.	TRM020 EQU \$		
1455	5563	21	AC	91	LXI H,DCCT ;SET DCCT ADDR		
1456	5566	7E	.	.	MOV A,M ;GET CONTROL WD		
1457	5567	E6	81	.	ANI DCCH+DCCA ;CLEAR NO PARITY		
1458	5569	77	.	.	MOV M,A ;SET NORMAL		
1459	556A	F6	01	.	ORI DCCA ;SET REQ TO SEND OFF		
1460	556C	67	.	.	MOV H,A		
1461	556D	3A	FC	FF	LDA KBDCSW ;TURN OFF REQUEST TO SEND		
1462	5570	E6	3E	.	ANI BAUDPT ;CLEAR H/F AND CA BITS		
1463	5572	.	.	.	TRM030 EQU \$		
1464	5572	84	.	.	ORA H		
1465	5573	32	40	81	STA IODCCT		
1466	5576	3E	01	.	MVI A,1 ;SET FOR 10 MSEC INTERVAL		
1467	5578	21	07	55	LXI H,BRK050 ;EXIT VIA "BRK050"		

13255
2648A MICROCODE LISTING 'DC16'

13255/90010
REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 41
=====
1468     557B     32  B3  91          STA  DCLAY
1469     557E     22  B1  91          SHLD DCTEX
1470     5581     21  6F  54          LXI  H,DCDCNT ;SET TO DELAY ROUTINE
1471     5584     22  B6  91          SHLD DCMVEC
1472     5587     C9  .   .          RET          ;RETURN TO TIMER INTERRUPT
=====
```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE  42
=====
1474      5588      . . .      ;
1475      5588      . . .      ; ENTMON - ENTER MONITOR MODE
1476      5588      . . .      ;
1477      5588      . . .      ENTMON EQU $
1478      5588      3E 02 .      MVI A,TRNMOD ;SET TRANSPARENT MODE
1479      558A      21 B8 91     LXI H,DCFLGS ;FLAG
1480      558D      F3 . .      DI
1481      558E      B6 . .      ORA M
1482      558F      77 . .      MOV M,A ;UPDATE FLAG SETTING
1483      5590      FB . .      EI
1484      5591      AF . .      XRA A
1485      5592      32 AE 91     STA FPMASK
1486      5595      C3 B3 54     JMP DCCTX1 ;RETURN
1487      5598      . . .      ;
1488      5598      . . .      ; RETNRM - RETURN TO NORMAL MODE
1489      5598      . . .      ;
1490      5598      . . .      RETNRM EQU $
1491      5598      3E FD .      MVI A,377Q-TRNMOD ;CLEAR TRANSPARENT MOD
1492      559A      21 B8 91     LXI H,DCFLGS
1493      559D      F3 . .      DI
1494      559E      A6 . .      ANA M
1495      559F      77 . .      MOV M,A ;UPDATE FLAG SETTINGS
1496      55A0      FB . .      EI
1497      55A1      E6 80 .      ANI FORPAR
1498      55A3      32 AE 91     STA FPMASK ;SET FP MASK
1499      55A6      C3 B3 54     JMP DCCTX1 ;RETURN
1500      55A9      . . .      ;
1501      55A9      . . .      ; FSTBIN - FAST BINARY ROUTINE GOTO 9600 BAUD
1502      55A9      . . .      ;
1503      55A9      . . .      FSTBIN EQU $
1504      55A9      3A FB FF     LDA KBJMPR ;READ JUMPER A-H
1505      55AC      E6 20 .      ANI FSTSND ;IS FAST SEND ENABLED?
1506      55AE      CA BA 55     JZ FST010 ;NO RETURN
1507      55B1      21 B8 91     LXI H,DCFLGS ;GET FLAGS
1508      55B4      F3 . .      DI
1509      55B5      7E . .      MOV A,M
1510      55B6      F6 10 .      ORI FB9600 ;SET FLAG FOR FAST BIN
1511      55B8      77 . .      MOV M,A ;STORE NEW FLAG
1512      55B9      FB . .      EI
1513      55BA      CD 68 51     FST010 CALL STRTBN ;SET BINARY OUTPUT MODE
1514      55BD      C3 B3 54     JMP DCCTX1 ;RETURN
=====

```

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 43
=====
1516     55C0      . . .      ;
1517     55C0      . . .      ; * * * * *
1518     55C0      . . .      ;
1519     55C0      . . .      ;   CHKDCM - WAIT FOR MONITOR TO CLEAR
1520     55C0      . . .      ;
1521     55C0      . . .      ;   ENTRY   DON'T CARE
1522     55C0      . . .      ;
1523     55C0      . . .      ;   EXIT    DCMVEC = DCMON
1524     55C0      . . .      ;           H-L DESTROYED
1525     55C0      . . .      ;           INTERRUPTS DISABLED
1526     55C0      . . .      CHKDCM EQU $
1527     55C0      21 2C 53    LXI H,DCMON ;GET NORMAL MONITOR ADDRESS
1528     55C3      F3 . .     DI          ;DISABLE INTERRUPTS
1529     55C4      3A B6 91    LDA DCMVEC  ;GET CURRENT MONITOR VECTOR
1530     55C7      BD . .     CMP L       ;IS IT THE NORMAL ROUTINE?
1531     55C8      C2 D0 55    JNZ CKD020 ;NO - TRY AGAIN
1532     55C8      3A B7 91    LDA DCMVEC+1
1533     55CE      BC . .     CMP H       ;DOES MSB MATCH?
1534     55CF      C8 . .     RZ          ;YES - RETURN
1535     55D0      . . .      CKD020 EQU $ ;NO - TRY AGAIN
1536     55D0      FB . .     EI          ;RE-ENABLE INTERRUPTS
1537     55D1      C3 C0 55    JMP CHKDCM  ;CHECK AGAIN
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1539	55D4	.	.	;	44
1540	55D4	.	.	*****	
1541	55D4	.	.	;	
1542	55D4	.	.	; SNDDC2 - SEND DC2	
1543	55D4	.	.	;	
1544	55D4	.	.	; ENTRY DON'T CARE	
1545	55D4	.	.	;	
1546	55D4	.	.	; EXIT A DESTROYED	
1547	55D4	.	.	;	
1548	55D4	.	.	;	
1549	55D4	.	.	SNDDC2 EQU \$	
1550	55D4	CD	AB	54 CALL CLBKTX ;GO TO CLEAR BLOCK TRIGGER	
1551	55D7	.	.	SNDD010 EQU \$	
1552	55D7	3E	12	. MVI A,DC2 ;SET DC2 AS CHAR	
1553	55D9	CD	9A	50 CALL PUTDC ;OUTPUT DC2	
1554	55DC	D8	.	. RC ;RETURN ON DC ERROR	
1555	55DD	C2	D7	55 JNZ SND010	
1556	55E0	3A	FB	FF LDA KBJMPR ;READ KB JUMPERS	
1557	55E3	2F	.	. CMA ;REVERSE SENSE OF BITS	
1558	55E4	E6	08	. ANI PAGSTR ;PAGE STRAP IN ?	
1559	55E6	C8	.	. RZ ;NO - RETURN	
1560	55E7	.	.	SNDD020 EQU \$	
1561	55E7	3E	0D	. MVI A,CR ;SET CR AS CHAR	
1562	55E9	CD	9A	50 CALL PUTDC ;OUTPUT CR	
1563	55EC	D8	.	. RC ;RETURN ON DC ERROR	
1564	55ED	C2	E7	55 JNZ SND020	
1565	55F0	3A	F3	FF LDA MDFLG2 ;READ MODE	
1566	55F3	E6	04	. ANI AUTOLF ;AUTO LF KEY DOWN?	
1567	55F5	C8	.	. RZ ;NO RETURN	
1568	55F6	.	.	SNDD030 EQU \$	
1569	55F6	3E	0A	. MVI A,LF ;SET LF AS CHAR	
1570	55F8	CD	9A	50 CALL PUTDC ;OUTPUT LF	
1571	55FB	D8	.	. RC ;RETURN ON DC ERROR	
1572	55FC	C2	F6	55 JNZ SND030	
1573	55FF	C9	.	. RET ;EXIT	

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 45
=====
1575      5600      . . .      ;
1576      5600      . . .      ;*****
1577      5600      . . .      ;
1578      5600      . . .      ; SNDTRM - SEND ED CHAR IF MAIN CHANNEL
1579      5600      . . .      ;
1580      5600      . . .      ;
1581      5600      . . .      SNDTRM EQU $
1582      5600      3A B8 91      LDA DCFLGS ;READ FLAGS
1583      5603      E6 21 .      ANI MCMOD+DCCA ;MAIN CHANNEL?
1584      5605      C2 B3 54      JNZ DCCTX1 ;NO RETURN
1585      5608      C3 3B 51      JMP PDC060 ;SEND ED CHAR
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1587	560B	.	.	;	46
1588	560B	.	.	; * * * * *	
1589	560B	.	.	;	
1590	560B	.	.	; DCTST - DATACOM SELF-TEST	
1591	560B	.	.	;	
1592	560B	.	.	; ENTRY DON'T CARE	
1593	560B	.	.	;	
1594	560B	.	.	; EXIT NC - SELF-TEST SUCCESSFUL	
1595	560B	.	.	; C - SELF-TEST FAILED	
1596	560B	.	.	; H,L = SELF-TEST MESSAGE	
1597	560B	.	.	; OTHER REGS. DESTROYED	
1598	560B	.	.	;	
1599	560B	.	.	DCTST EQU \$	
1600	560B	.	.	;*****	
1601	560B	.	.	; DISALLOW DATACOM *	
1602	560B	.	.	; INTERRUPTS *	
1603	560B	.	.	;*****	
1604	560B	3A	F9 FF	LDA KBJMP3	
1605	560E	E6	10 .	ANI NODCST	
1606	5610	C2	51 57	JNZ SFT600	
1607	5613	21	F5 FF	LXI H,PRCTL ;PROCESSOR CONTROL FLAG	
1608	5616	F3	. .	DI	
1609	5617	7E	. .	MOV A,M	
1610	5618	F6	10 .	ORI DCIOFF ;DON'T LET TIMER ROUTINE	
1611	561A	77	. .	MOV M,A ;REENABLE DATACOM INT.	
1612	561B	D3	70 .	OUT PROCSR ;DISABLE DATACOM INT.	
1613	561D	FB	. .	EI	
1614	561E	.	.	;	
1615	561E	21	5A 57	LXI H,TMOUT ;GIVE CONTROL TO TIMEOUT	
1616	5621	22	B6 91	SHLD DCMVEC ;ROUTINE ON TIMER INTS.	
1617	5624	.	.	;*****	
1618	5624	.	.	; DATACOM *	
1619	5624	.	.	; ALIVE? *	
1620	5624	.	.	;*****	
1621	5624	97	. .	SUB A ;RAISE REQUEST TO SEND	
1622	5625	32	40 81	STA IODCCT	
1623	5628	06	20 .	MVI B,DCCB ;CHECK FOR CLEAR TO SEND	
1624	562A	48	. .	MOV C,B	
1625	562B	CD	5F 57	CALL STATCH ;CHECK DATA COM STATUS	
1626	562E	21	D0 57	LXI H,STCAB ;ERROR MESSAGE POINTER	
1627	5631	CA	22 57	JZ SFT400 ;INDICATE FAILURE	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 47
1629	5634	.	.	*****	
1630	5634	.	.	; CHECK DATA *	
1631	5634	.	.	; SEND RECEIVE *	
1632	5634	.	.	; LOOP *	
1633	5634	.	.	*****	
1634	5634	3E	20	MVI A,DCNP ;SET INITIAL PARITY - NONE	
1635	5636	.	.	SFT130 EQU \$	
1636	5636	32	B0 91	STA TPARIT	
1637	5639	06	0E	MVI B,B9600 ;9600 BAUD	
1638	563B	80	.	ADD B ;COMBINE BAUD RATE AND PARIT	
1639	563C	32	40 81	STA IODCCT ;SET THEM	
1640	563F	3A	00 81	LDA IODCDI ;CLEAR DATA IN REGISTER	
1641	5642	16	00	MVI D,0 ;INITIALIZE TO FIRST CHAR	
1642	5644	.	.	; SEND CHARACTER	
1643	5644	.	.	SFT170 EQU \$	
1644	5644	06	02	MVI B,DCTBE ;WAIT TILL XMIT BUFER MT	
1645	5646	0E	00	MVI C,0	
1646	5648	CD	5F 57	CALL STATCH	
1647	564B	21	AB 57	LXI H,STDATA ;IN CASE OF FAIL	
1648	564E	CA	22 57	JZ SFT400 ;FAILURE	
1649	5651	7A	.	MOV A,D	
1650	5652	32	60 81	STA IODCDO ;SEND DATA	
1651	5655	06	01	MVI B,DCDP ;WAIT UNTIL DATA PRESENT	
1652	5657	0E	00	MVI C,0	
1653	5659	CD	5F 57	CALL STATCH	
1654	565C	CA	22 57	JZ SFT400 ;FAILED-NO CHAR REC'D	
1655	565F	3A	00 81	LDA IODCDI ;GET CHARACTER	
1656	5662	47	.	MOV B,A ;SAVE	
1657	5663	.	.	; CHECK PARITY	
1658	5663	.	.	SFT200 EQU \$	
1659	5663	3A	20 81	LDA IODCST ;DATACOM STATUS	
1660	5666	4F	.	MOV C,A ;SAVE	
1661	5667	E6	08	ANI DCPE ;PARITY ERROR BIT SET?	
1662	5669	21	B9 57	LXI H,STPARE ;PARITY ERROR MSG PTR	
1663	566C	C2	22 57	JNZ SFT400 ;YES, ERROR	
1664	566F	.	.	; OVERRUN ERROR?	
1665	566F	79	.	MOV A,C ;STATUS	
1666	5670	E6	04	ANI DCOE ;CHECK BIT	
1667	5672	21	C4 57	LXI H,STOVRE ;MSG POINTER	
1668	5675	C2	22 57	JNZ SFT400 ;ERROR	
1669	5678	.	.	; CHECK CHARACTER	
1670	5678	78	.	MOV A,B ;DATACOM CHARACTER	
1671	5679	BA	.	CMP D ;SAME AS SENT?	
1672	567A	21	AB 57	LXI H,STDATA ;ERROR POINTER	
1673	567D	C2	22 57	JNZ SFT400	
1674	5680	.	.	; DO NEXT CHARACTER	
1675	5680	14	.	INR D ;BUMP TO NEXT CHARACTER	
1676	5681	CA	8F 56	JZ SFT260 ;WAS 8 BIT TEST, DONE	
1677	5684	F2	44 56	JP SFT170 ;.LE. 177B	
1678	5687	3A	B0 91	LDA TPARIT ;NO PARITY?	

13255

2648A MICROCODE LISTING 'DC16'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 48
=====
```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE
1679	568A	FE	20 .	CPI DCNP	
1680	568C	CA	44 56	JZ SFT170 ;YES, USE 8 BITS	
1681	568F	.	.	;	
1682	568F	.	.	; DO NEXT PARITY SETTING	
1683	568F	.	.	;	
1684	568F	.	.	SFT260 EQU \$	
1685	568F	3A	80 91	LDA TPARIT	
1686	5692	FE	20 .	CPI DCNP	
1687	5694	CA	A1 56	JZ SFT280 ;NO PARITY, USE ODD NEXT	
1688	5697	FE	00 .	CPI DCOP ;ODD PARITY	
1689	5699	C2	A6 56	JNZ SFT300 ;NO, MUST BE DONE	
1690	569C	3E	10 .	MVI A,DCEP ;DO EVEN PARITY	
1691	569E	C3	36 56	JMP SFT130	
1692	56A1	.	.	SFT280 EQU \$	
1693	56A1	3E	00 .	MVI A,DCOP ;ODD PARITY	
1694	56A3	C3	36 56	JMP SFT130	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 49
=====
1696     56A6      . . .      ;*****
1697     56A6      . . .      ; TEST OTHER *
1698     56A6      . . .      ; BAUD RATES *
1699     56A6      . . .      ;*****
1700     56A6      . . .      SFT300 EQU $
1701     56A6      16 07 .      MVI D,7          ;BAUD COUNTER
1702     56A8      . . .      SFT320 EQU $
1703     56A8      15 . .      DCR D            ;DO NEXT BAUD RATE
1704     56A9      CA E1 56      JZ SFT340        ;ALL BAUD RATES TESTED
1705     56AC      7A . .      MOV A,D          ;FORMAT CONTROL WORD
1706     56AD      07 . .      RLC              ;SHIFT TO BITS 1-3
1707     56AE      06 20 .      MVI B,DCNP      ;NO PARITY
1708     56B0      80 . .      ADD B
1709     56B1      32 40 81      STA IODCCT       ;SET NO PARITY AND BAUD
1710     56B4      97 . .      SUB A
1711     56B5      32 60 81      STA IODCDO       ;SEND NULL
1712     56B8      06 01 .      MVI B,DCDP
1713     56BA      0E 00 .      MVI C,0
1714     56BC      CD 5F 57      CALL STATCH      ;WAIT FOR DATA
1715     56BF      21 AB 57      LXI H,STDATA    ;ERROR MSG
1716     56C2      CA 22 57      JZ SFT400        ;NO DATA RECEIVED
1717     56C5      3A 00 81      LDA IODCDI       ;GET DATA
1718     56C8      B7 . .      ORA A            ;IS IT ZERO?
1719     56C9      C2 22 57      JNZ SFT400      ;NO, ERROR
1720     56CC      3E FF .      MVI A,377Q      ;SEND ALL ONES
1721     56CE      32 60 81      STA IODCDO
1722     56D1      CD 5F 57      CALL STATCH
1723     56D4      CA 22 57      JZ SFT400        ;ERROR
1724     56D7      3A 00 81      LDA IODCDI       ;CHECK DATA
1725     56DA      3C . .      INR A
1726     56DB      C2 22 57      JNZ SFT400      ;NOT 377, ERROR
1727     56DE      C3 A8 56      JMP SFT320       ;DO NEXT BAUD RATE
=====

```

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE 50
1729	56E1	.	.	*****	
1730	56E1	.	.	; TEST *	
1731	56E1	.	.	; LINES *	
1732	56E1	.	.	*****	
1733	56E1	.	.	SFT340 EQU \$	
1734	56E1	11	EF 57	LXI D,LINTBL ;TABLE OF PATTERNS	
1735	56E4	.	.	SFT360 EQU \$	
1736	56E4	1A	.	LDAX D ;GET DATACOM CONTROL WORD	
1737	56E5	3C	.	INR A ;END?	
1738	56E6	CA	1B 57	JZ SFT380 ;DONE, ALL TESTS PASSED	
1739	56E9	3D	.	DCR A ;RESTORE A	
1740	56EA	.	.	; SET DATACOM CONTROL	
1741	56EA	32	40 81	STA IODCCT	
1742	56ED	06	20 .	MVI B,DCCB ;CHECK CB	
1743	56EF	13	.	INX D ;BUMP TO NEXT TABLE ENTRY	
1744	56F0	1A	.	LDAX D	
1745	56F1	4F	.	MOV C,A	
1746	56F2	CD	5F 57	CALL STATCH	
1747	56F5	21	DA 57	LXI H,STCB	
1748	56F8	CA	22 57	JZ SFT400 ;CB ERROR	
1749	56FB	06	10 .	MVI B,DCCF ;CHECK CF	
1750	56FD	13	.	INX D	
1751	56FE	1A	.	LDAX D	
1752	56FF	4F	.	MOV C,A	
1753	5700	CD	5F 57	CALL STATCH	
1754	5703	21	E1 57	LXI H,STCACF	
1755	5706	CA	22 57	JZ SFT400 ;ERROR	
1756	5709	06	40 .	MVI B,DCSB ;CHECK SB	
1757	570B	13	.	INX D	
1758	570C	1A	.	LDAX D	
1759	570D	4F	.	MOV C,A	
1760	570E	CD	5F 57	CALL STATCH	
1761	5711	21	E8 57	LXI H,STBKSB	
1762	5714	CA	22 57	JZ SFT400 ;ERROR	
1763	5717	.	.	;	
1764	5717	13	.	INX D ;BUMP TO NEXT TEST SET	
1765	5718	C3	E4 56	JMP SFT360	

```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                     PAGE 51
=====
1767      5718      .      .      .      ;*****
1768      5718      .      .      .      ; REPORT FINAL *
1769      5718      .      .      .      ; STATUS BACK *
1770      5718      .      .      .      ;*****
1771      5718      .      .      .      ;
1772      5718      .      .      .      ; GOOD STATUS
1773      5718      .      .      .      ;
1774      5718      .      .      .      SFT380 EQU $
1775      5718      21      8E      57      LXI H,STGOOD ;GOOD STATUS
1776      571E      A7      .      .      ANA A ;CLEAR CARRY
1777      571F      C3      29      57      JMP SFT520 ;FINISH
1778      5722      .      .      .      ;
1779      5722      .      .      .      ; ERRORS
1780      5722      .      .      .      ;
1781      5722      .      .      .      SFT400 EQU $
1782      5722      22      EB      FF      SHLD MSGPT4 ;SET FAIL MSG
1783      5725      21      A4      57      LXI H,STFAIL ;FAIL LITERAL
1784      5728      37      .      .      STC ;SET CARRY TO INDICATE ERROR
1785      5729      .      .      .      ;*****
1786      5729      .      .      .      ; RETURN *
1787      5729      .      .      .      ;*****
1788      5729      .      .      .      SFT520 EQU $
1789      5729      F5      .      .      PUSH PSW ;SAVE FLAGS
1790      572A      22      ED      FF      SHLD MSGPT3 ;SET MESSAGE TYPE
1791      572D      06      01      .      MVI B,DCDP ;CLEAR CHARACTER INPUT BUFFE
1792      572F      0E      00      .      MVI C,0
1793      5731      CD      5F      57      CALL STATCH
1794      5734      3A      00      81      LDA IODCDI
1795      5737      .      .      .      ;
1796      5737      21      2C      53      LXI H,DCMON ;RESET DC MONITOR ADDRESS
1797      573A      22      B6      91      SHLD DCMVEC
1798      573D      21      F5      FF      LXI H,PRCCTL ;PRECESSOR CONTROL FLAG
1799      5740      F3      .      .      DI
1800      5741      7E      .      .      MOV A,M
1801      5742      E6      EF      .      ANI 377Q-DCIOFF ;ENABLE DATACOM INTS.
1802      5744      77      .      .      MOV M,A
1803      5745      FB      .      .      EI
1804      5746      F1      .      .      POP PSW
1805      5747      21      83      57      SFT530 LXI H,STSFTS
1806      574A      22      EF      FF      SHLD MSGPT2
1807      574D      21      72      57      LXI H,DCTYPE ;SET DATACOM TYPE MESSAGE
1808      5750      C9      .      .      RET
1809      5751      21      91      57      SFT600 LXI H,STDISA
1810      5754      22      ED      FF      SHLD MSGPT3
1811      5757      C3      47      57      JMP SFT530
=====

```

13255

2648A MICROCODE LISTING 'DC16'

13255/90010

REV 04/17/78

```
=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS                                PAGE 52
=====
1813     575A      . . .      ;*****
1814     575A      . . .      ; TIMING ROUTINE                               *
1815     575A      . . .      ; THIS IS CALLED ON                           *
1816     575A      . . .      ; EVERY TIMER INTERRUPT                       *
1817     575A      . . .      ; AND DECREMENTS A COUNTER                   *
1818     575A      . . .      ;                                             *
1819     575A      . . .      ;          ENTRY DON'T CARE                   *
1820     575A      . . .      ;          EXIT H,L,FLAGS                     *
1821     575A      . . .      ;          CHANGED                            *
1822     575A      . . .      ;*****
1823     575A      . . .      TMOUT EQU $
1824     575A      21 AF 91   LXI H,TMOCNT ;COUNTER
1825     575D      35 . .     DCR M
1826     575E      C9 . .     RET
=====
```

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE 53
1828	575F	. . .	;*****	
1829	575F	. . .	; CHECK DATACOM STATUS	*
1830	575F	. . .	; AND WATCH FOR TIMEOUT	*
1831	575F	. . .	;	*
1832	575F	. . .	; ENTRY B= BIT IN STATUS	*
1833	575F	. . .	; TO BE EXAMINED	*
1834	575F	. . .	;	*
1835	575F	. . .	; C= B IF WE WANT BIT	*
1836	575F	. . .	; TO TURN ZERO	*
1837	575F	. . .	; C= 0 IF WE WANT BIT	*
1838	575F	. . .	; TO EQUAL ONE	*
1839	575F	. . .	; --SO EXCLUSIVE OR OF	*
1840	575F	. . .	; STATUS AND C REG IS	*
1841	575F	. . .	; NON ZERO WHEN BIT	*
1842	575F	. . .	; BECOMES THE VALUE	*
1843	575F	. . .	; WE ARE LOOKING FOR	*
1844	575F	. . .	;	*
1845	575F	. . .	; EXIT A= 0 IF BIT DID NOT	*
1846	575F	. . .	; TURN BY 400 MS	*
1847	575F	. . .	; A .NE. 0 OTHERWISE	*
1848	575F	. . .	; OTHER REGS. SAVED	*
1849	575F	. . .	;*****	
1850	575F	. . .	STATCH EQU \$	
1851	575F	3E 14 .	MVI A,20	
1852	5761	32 AF 91	STA TMCNT ;INITIALIZE COUNTER	
1853	5764	. . .	; EXAMINE STATUS	
1854	5764	. . .	STW100 EQU \$	
1855	5764	3A 20 81	LDA IODCST	
1856	5767	A0 . .	ANA B ;GET BIT DESIRED	
1857	5768	A9 . .	XRA C ;BIT IN RIGHT STATE?	
1858	5769	C0 . .	RNZ	
1859	576A	. . .	; TIME OUT YET?	
1860	576A	3A AF 91	LDA TMCNT ;TIME OUT COUNTER	
1861	576D	B7 . .	ORA A ;ZERO?	
1862	576E	C2 64 57	JNZ STW100 ;NO	
1863	5771	C9 . .	RET ;YES, NEVER FOUND RIGHT STAT	

ITEM	LOC	OBJECT	CODE	SOURCE STATEMENTS	PAGE	54
1865	5772	.	.	. ;*****		
1866	5772	.	.	. ; SELF TEST *		
1867	5772	.	.	. ; LITERALS *		
1868	5772	.	.	. ;*****		
1869	5772	.	.	. DCTYPE EQU \$		
1870	5772	42	41	53 DB 'BASIC DATA COMM'		
1871	5781	CC	00	. DB EOL,0		
1872	5783	.	.	. STSFTS EQU \$		
1873	5783	53	45	4C DB 'SELF TEST ',0		
1874	578E	.	.	. STGOOD EQU \$		
1875	578E	4F	4B	CE DB 'OK',EOP		
1876	5791	.	.	. STDISA EQU \$		
1877	5791	45	52	52 DB 'ERROR 0 (DISABLED)',EOP		
1878	57A4	.	.	. STFAIL EQU \$		
1879	57A4	45	52	52 DB 'ERROR ',0		
1880	57AB	.	.	. STDATA EQU \$		
1881	57AB	33	20	28 DB '3 (LOST CHAR)',EOP		
1882	57B9	.	.	. STPARE EQU \$		
1883	57B9	32	20	28 DB '2 (PARITY)',EOP		
1884	57C4	.	.	. STOVRE EQU \$		
1885	57C4	33	20	28 DB '3 (OVERRUN)',EOP		
1886	57D0	.	.	. STCACB EQU \$		
1887	57D0	31	20	28 DB '1 (NO CB)',EOP		
1888	57DA	.	.	. STCB EQU \$		
1889	57DA	34	20	28 DB '4 (CB)',EOP		
1890	57E1	.	.	. STCACF EQU \$		
1891	57E1	34	20	28 DB '4 (CF)',EOP		
1892	57E8	.	.	. STBKSB EQU \$		
1893	57E8	35	20	28 DB '5 (SB)',EOP		


```

=====
ITEM      LOC      OBJECT CODE  SOURCE STATEMENTS
=====
1895      57EF      . . .      ;*****
1896      57EF      . . .      ; DATACOM LINE TEST PATTERNS *
1897      57EF      . . .      ;
1898      57EF      . . .      ; 4 BYTES/ENTRY-TEST *
1899      57EF      . . .      ; 1ST = DATACOM CONTROL BYTE *
1900      57EF      . . .      ; 2ND = CB WORD *
1901      57EF      . . .      ; 3RD = CF WORD *
1902      57EF      . . .      ; 4TH = SB WORD *
1903      57EF      . . .      ; THESE LAST 3 BYTES GO TO *
1904      57EF      . . .      ; C REG WHEN STATCH CALLED *
1905      57EF      . . .      ;*****
1906      57EF      . . .      LINTBL EQU $
1907      57EF      00 . . .      DB 0 ;NOT CA = 0, BRK = 0
1908      57F0      20 . . .      DB DCCB ;NOT CB = 0
1909      57F1      10 . . .      DB DCCF ;NOT CF = 0
1910      57F2      00 . . .      DB 0 ;NOT SB = 1
1911      57F3      . . .      ;
1912      57F3      01 . . .      DB 1 ;NOT CA = 1, BRK = 0
1913      57F4      00 . . .      DB 0 ;NOT CB = 1
1914      57F5      00 . . .      DB 0 ;NOT CF = 1
1915      57F6      40 . . .      DB DCSB ;NOT SB = 0
1916      57F7      . . .      ;
1917      57F7      41 . . .      DB 101Q ;CA = 1, BRK = 1
1918      57F8      00 . . .      DB 0 ;NOT CB = 1
1919      57F9      00 . . .      DB 0 ;NOT CF = 1
1920      57FA      00 . . .      DB 0 ;NOT SB = 1
1921      57FB      . . .      ;
1922      57FB      FF . . .      DB 377Q ;END OF TABLE
=====

```

13255

2648A MICROCODE LISTING 'DC16'

13255/90010

REV 04/17/78

=====

ITEM	LOC	OBJECT CODE	SOURCE STATEMENTS	PAGE
------	-----	-------------	-------------------	------

=====

1924	57FC	.	.	END
------	------	---	---	-----

0 ERRORS FOUND IN ASSEMBLY CODE .

13255
2648A MICROCODE LISTING 'DC16'
SYMBOL VALUE REFERENCED ON

```

=====
ACAN      0018      298
ACK       0006      296, 606, 947
ADEL     007F      300, 446, 652, 657
ANULL    0000      288
AUTOLF   0004      137, 635, 642, 1566
AUTTRM   0001      68
B9600    000E      257, 566, 1046, 1637
BAUDPT   003E      265, 569, 757, 1049, 1249, 1375, 1401, 1462
BIGBUF   0800      19, 700
BINMOD   0004      220, 441, 591, 797, 808, 885, 958, 987
BLKMDE   0002      136, 625
BLKTRG   0001      92, 1136, 1183, 1323, 1340
BLKTRM   5004      385
BREAK    54DA      1369, 1307
BRK005   54E8      1376
BRK010   54F1      1381, 1416, 1442
BRK050   5507      1397, 1379, 1467
BRK100   5510      1402
CAPSLK   0001      135
CBKSTP   0004      356, 1162
CDOFF    0008      263, 1413
CFSTRP   0008      359, 900, 1078
CHEKCC   0040      85, 726, 1059
CHKDCM   55C0      1526, 1370, 1386, 1412, 1438, 1448, 1537
CKD020   55D0      1535, 1531
CLBKTX   54AB      1320, 1550
CLBLTR   54AC      1322, 1302
CLRTRG   0000      413
CLRTRM   0002      69
CMBASE   00FF      153, 154
CMFLGS   FFF8      162, 163, 1134, 1180, 1324, 1341
CMSTOR   FF00      154
COMMON   FFFF      152, 153, 156
CONDIS   0001      39
CR        000D      294, 613, 1561
CTIJMP   FFE0      177, 178
CTIVEC   FFE1      176, 177
DC2       0012      297, 622, 1552
DC2SND   0080      58
DCBASE   0091      312
DCBEND   91AA      329, 519, 702, 714, 716
DCBFBG   91B9      316, 317, 513, 516, 719, 769, 1205
DCBPTR   91BD      314, 315, 487, 490, 771, 837, 866, 1207
DCCA     0001      218, 808, 809, 944, 1126, 1185, 1192, 1244, 1338, 1457,
          1459, 1583
DCCB     0020      210, 1065, 1106, 1623, 1742, 1908
DCCC     0080      213, 1062
DCCF     0010      209, 1082, 1749, 1909
DCCH     0080      264, 736, 753, 1130, 1457
DCCT     91AC      327, 329, 551, 682, 684, 740, 752, 754, 1042, 1128,
          1190, 1243, 1371, 1398, 1455
DCCTAB   5491      1301, 1290
DCCTL    547B      1281, 401, 767

```

13255
 2648A MICROCODE LISTING 'DC16'
 SYMBOL VALUE REFERENCED ON

13255/90010
 REV 04/17/78

```

=====
DCCTL2  8160    248, 1394, 1413
DCCTX1  54B3    1327, 1364, 1339, 1344, 1350, 1360, 1388, 1486, 1499, 1514,
          1584
DCDCNT  546F    1256, 1241, 1384, 1470
DCDLAY  91B3    321, 322, 1239, 1257, 1382, 1468
DCDP    0001    200, 434, 848, 1651, 1712, 1791
DCEP    0010    260, 1690
DCFLGS  9188    317, 318, 439, 553, 589, 751, 794, 819, 883, 902,
          943, 952, 957, 959, 985, 1044, 1086, 1123, 1147, 1186,
          1219, 1337, 1423, 1479, 1492, 1507, 1582
DCINTR  5026    429, 850
DCIOFF  0010    116, 1610, 1801
DCJMK2  5006    393
DCJMP0  0080    73
DCJMP1  0001    77
DCJMP2  0002    78
DCJMP3  0004    79
DCJMP4  0008    80
DCJMSK  5005    386
DCM002  533F    1050, 1047
DCM005  535D    1064, 1060
DCM010  5360    1066, 1063
DCM030  5375    1085, 910, 918
DCM040  5384    1095
DCM050  538C    1102, 1097
DCM055  53A6    1114, 1107
DCM060  53AE    1122, 1110, 1113
DCM070  53D6    1146
DCM075  53DC    1150, 1153, 1154
DCM100  53E8    1161, 1071
DCM105  53F7    1172, 1175, 1176
DCM110  5404    1179, 662, 1117, 1171
DCM150  5424    1201, 1196
DCM160  5448    1218, 1215
DCMERR  0001    101, 469, 558, 873
DCMJMP  91B5    319, 320, 400, 761
DCMON   532C    1041, 578, 762, 1132, 1202, 1405, 1527, 1796
DCMVEC  91B6    318, 319, 763, 1133, 1203, 1242, 1261, 1385, 1406, 1471,
          1529, 1532, 1616, 1797
DCNP    0020    261, 566, 683, 1046, 1130, 1634, 1679, 1686, 1707
DCOE    0004    202, 460, 1666
DCOP    0000    259, 1688, 1693
DCPE    0008    203, 453, 1661
DCSA    0040    258, 1372
DCSB    0040    211, 1112, 1166, 1756, 1915
DCSPTR  91BB    315, 316, 476, 485, 770, 839, 842, 1206
DCSTAT  91BF    313, 314, 765, 1054, 1104, 1210
DCSTOR  91C0    311, 312, 313
DCT010  5465    1247
DCTBE   0002    201, 574, 680, 1436, 1446, 1644
DCTEX   91B1    322, 323, 1240, 1260, 1383, 1469
DCTST   560B    1599, 402
DCTURN  5452    1238, 1098, 1197
  
```

13255

2648A MICROCODE LISTING 'DC16'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
DCTYPE      5772      1869, 1807
DECRDX      000A      143
DEFSKY      0008      95
DISCNT      0006      419
DISPST      FFFE      156, 157
DSC100      5504      1393, 1415
DSCNCT      551B      1411, 1308
DSPFNC      0001      124
EDIT        0010      128
EDTWRP      0008      71
ENDBLK      0007      420
ENDCHR      91AD      326, 327, 599, 645, 733, 893, 921, 1173
ENQ         0005      295, 930
ENTMON      5588      1477, 1310
EOL         00CC      306, 1871
EOP         00CE      307, 1875, 1877, 1881, 1883, 1885, 1887, 1889, 1891, 1893
EOT         0004      292, 731
ERRFLG      FFF7      163, 164, 467, 556, 872
ESCFLG      FFD1      189, 192
ETX         0003      291, 729
ETXSTP      0002      347, 728
FB9600      0010      222, 565, 1045, 1426, 1510
FORGN       0080      131
FORMAT      0008      127
FORPAR      0080      225, 456, 799, 1429, 1497
FPMASK      91AE      325, 326, 586, 686, 749, 800, 962, 1430, 1485, 1498
FRCPTY      0080      87, 748
FRCRST      0004      94
FST010      558A      1513, 1506
FSTBIN      55A9      1503, 1312
FSTRAM      9100      26, 149, 311
FSTSND      0020      53, 1505
FULDUP      0080      30
GDC001      521A      802, 990
GDC002      5249      828, 810, 813, 815, 816, 818, 849, 908, 1313, 1314
GDC004      524E      833, 814, 942
GDC005      524F      835, 806, 852
GDC007      5270      857, 841, 844
GDC008      5273      864
GDC010      528B      882, 871, 874
GDC012      528F      909, 901
GDC014      52C8      916, 898
GDC015      52D3      925, 892, 895
GDC020      52E3      941, 945
GDC030      52EE      946, 950, 951
GDC050      5311      968, 886, 928, 934
GETBIN      5315      983, 405
GETDC       520B      792, 403, 995
GOBIN       0008      221, 565, 954, 958, 991
GPASYC      0080      331, 722, 1432
HNDSHK      0040      55
IDC010      518B      701, 699
INI2DC      5191      710, 399

```

13255

2648A MICROCODE LISTING 'DC16'

13255/90010

REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
INITDC  517D    692,  398
INSCHR  0002    125
INSWRP  0002     93
INT020  5059    458,  442,  454
INT040  5080    493,  489
INT050  505F    465,  457,  492
INT100  5068    475,  461
INT130  506E    483
INT160  5081    499,  445,  451
INT170  5082    501,  435
INT180  5086    512,  480,  861
INTFLG  FFF6    164,  165
INTVEC  9165    149,  150
INVRS   0082    305
IOBASE  0080    230,  235,  240
IOCSGN  FFDD    179,  180
IODATA  FFDE    178,  179
IODC    8100    240,  244,  245,  246,  247,  248,  252,  253
IODCCT  8140    252,  572,  759,  1051,  1251,  1378,  1404,  1465,  1622,  1639,
                1709,  1741
IODCDI  8100    244,  437,  1640,  1655,  1717,  1724,  1794
IODCDO  8160    253,  588,  1650,  1711,  1721
IODCPC  8140    247,  721,  1431
IODCS2  8121    246,  1061,  1435
IODCST  8120    245,  432,  573,  679,  764,  847,  1052,  1103,  1209,  1445,
                1659,  1855
IOKB    8300    235,  236
IOKBLD  8300    236
IOPSGN  FFDC    180,  181
JMP     00C3    284,  760
KBDCSW  FFFC    158,  159,  568,  756,  1048,  1248,  1374,  1400,  1461
KBJMP2  FFFA    160,  161,  697,  1108
KBJMP3  FFF9    161,  162,  723,  727,  811,  899,  1057,  1139,  1211,  1356,
                1604
KBJMPR  FFFB    159,  160,  608,  1504,  1556
LDRCHK  0004    103
LF      000A    293,  615,  1569
LFPOS   0010     47
LINTBL  57EF    1906,  1734
LINWRP  0004     43
MCMOD   0020    223,  591,  746,  891,  1583
MDFLG1  FFF4    166,  167
MDFLG2  FFF3    167,  168,  624,  634,  641,  1565
MEMLOK  0004    126
MNCHAN  0003    351,  743,  1141,  1170,  1213
MSGPT1  FFF1    168,  169
MSGPT2  FFEF    169,  170,  1806
MSGPT3  FFED    170,  171,  1790,  1810
MSGPT4  FFE8    171,  172,  1782
MSGPT5  FFE9    172,  173
MSGPT6  FFE7    173,  174
MSGPT7  FFE5    174,  175
MSGPT8  FFE3    175,  176

```

13255
2648A MICROCODE LISTING 'DC16'

SYMBOL	VALUE	REFERENCED ON
NODCST	0010	81, 1605
NORMAL	0080	304
NOTEST	0004	70
NRMBUF	0060	18, 696
OCTRDY	0008	144
PAGSTR	0008	45, 609, 1558
PARM1	FFDB	181, 182
PARM2	FFDA	182, 183
PARM3	FFD9	183, 184
PARM4	FFD8	184, 185
PARM5	FFD7	185, 186
PARM6	FFD5	186, 187
PDC004	50B2	564, 555
PDC005	50BE	570, 567
PDC020	50D5	585, 575
PDC025	50E5	593, 604
PDC030	50E8	598, 592
PDC035	5103	611, 626
PDC040	5111	620, 610
PDC050	5128	633, 614
PDC052	5130	637, 643
PDC055	5133	640, 616
PDC060	5138	644, 607, 623, 629, 636, 647, 648, 1585
PDC070	5149	651, 601, 655, 656
PDC080	5155	657, 659, 660
PJMPR	0020	16, 698
POLL	0040	118
PRCCTL	FFF5	165, 166, 1607, 1798
PRNTAL	0010	72
PROCSR	0070	231, 1612
PROMPT	000D	423, 1282
PUTBRK	0005	418
PUTDC	509A	548, 404, 646, 654, 658, 949, 1152, 1174, 1553, 1562, 1570
QJMPR	0040	17, 698
RADIX	FFD4	187, 188
RCVMDE	0020	97
RECORD	0040	130
RECSEP	5003	384
REMOTE	0008	138
REMSET	0010	96
RETNRM	5598	1490, 1311
RNGTA	FFD2	188, 189
RS	001E	299, 628
RSETDC	0002	415
RST005	51AD	728, 725
RST006	51B6	732, 730
RST007	51C2	739, 737
RST010	51CF	747, 745
RST020	51D5	750
RSTDC1	5201	768
RSTDCB	519C	720, 1349
RSTDCM	54C8	1348, 1304

13255

2648A MICROCODE LISTING 'DC16'

13255/90010
REV 04/17/78

SYMBOL VALUE REFERENCED ON

```

=====
RSTSRM  51C6    742, 1358
RSTTMR  FFD0    192
SBSTRP  0080    340, 1109
SCLD    0040    14, 529
SCNVEC  9168    150
SELECT  0020    129
SETCH   0020    83, 735
SETLCL  0004    417
SETMON  0008    421
SETNRM  0009    422
SETREM  0003    416
SETROM  0000    120
SETTRG  0001    414
SFT130  5636    1635, 1691, 1694
SFT170  5644    1643, 1677, 1680
SFT200  5663    1658
SFT260  568F    1684, 1676
SFT280  56A1    1692, 1687
SFT300  56A6    1700, 1689
SFT320  56A8    1702, 1727
SFT340  56E1    1733, 1704
SFT360  56E4    1735, 1765
SFT380  571B    1774, 1738
SFT400  5722    1781, 1627, 1648, 1654, 1663, 1668, 1673, 1716, 1719, 1723,
              1726, 1748, 1755, 1762
SFT520  5729    1788, 1777
SFT530  5747    1805, 1811
SFT600  5751    1809, 1606
SND010  55D7    1551, 1555
SND020  55E7    1560, 1564
SND030  55F6    1568, 1572
SNDDC2  55D4    1549, 1283
SNDTRM  5600    1581, 1309
SOH     0001    289
SPECHO  0040    224, 804, 805, 822, 905, 927, 1089, 1222
SPLDIS  0002    41
STATCH  575F    1850, 1625, 1646, 1653, 1714, 1722, 1746, 1753, 1760, 1793
STB010  5168    678, 681
STBKSB  57E8    1892, 1761
STBLTR  54B6    1336, 1303
STCACB  57D0    1886, 1626
STCACF  57E1    1890, 1754
STCB    57DA    1888, 1747
STDATA  57AB    1880, 1647, 1672, 1715
STDISA  5791    1876, 1809
STFAIL  57A4    1878, 1783
STGOOD  578E    1874, 1775
STLOCL  54B3    1364, 1306
STOVRE  57C4    1884, 1667
STPARE  57B9    1882, 1662
STRMTE  54CE    1354, 1305
STRTBN  5168    677, 1513
STSFTS  5783    1872, 1805

```


13255
2648A MICROCODE LISTING 'DC16'
SYMBOL VALUE REFERENCED ON

```

=====
STW100  5764  1854, 1862
STX      0002  290, 817, 1151
STXSTP   0001  344, 812, 1144, 1216
TESTOK   0002  102
TMIACK   0000  112
TMIEN    0002  115
TMIOFF   0020  117
TMOCNT   91AF  324, 325, 1824, 1852, 1860
TMOU     575A  1823, 1615
TMRINT   0003  107
TMRUN    0001  114
TPARIT   9180  323, 324, 1636, 1678, 1685
TRIGGR   5002  383
TRM005   5538  1431
TRM010   5540  1434, 1437
TRM012   5540  1440, 1450
TRM015   5553  1444, 1433, 1447
TRM020   5563  1454, 1441
TRM030   5572  1463
TRMBIN   5529  1421, 409
TRMTYP   FFFD  157, 158
TRNMOD   0002  219, 441, 603, 804, 897, 927, 1148, 1478, 1491
UNSCLO   0001  15, 529
WBSR     0020  139
XMTDLY   91B4  320, 321, 1093, 1115
ZBELL    4814  277, 278
ZCHKTK   5094  527, 449, 933
ZCLMD1   4811  276, 277
ZCLXMT   481A  279, 1067
ZGETKY   4805  272, 273
ZINIKB   4802  271, 272
ZKBBAS   4800  270, 271
ZKBCTL   4808  273, 274
ZKBMON   480B  274, 275
ZSTMD1   480E  275, 276
ZSTXMT   4817  278, 279, 1066
ZTKFLG   90AD  13, 528
=====

```

344 SYMBOLS, 1002 REFERENCES, 24 WORK TRACKS

