96729000

CONTROL DATA CORPORATION

CDC® BREAKPOINT CONTROLLER AND BREAKPOINT PANEL

DT195-A (FC402-A, DT120-A)

GENERAL DESCRIPTION
OPERATION AND PROGRAMMING
INSTALLATION AND CHECKOUT
THEORY OF OPERATION
DIAGRAMS
MAINTENANCE

HARDWARE MAINTENANCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
01	Preliminary release.
(11/75)	
A	Manual released.
(5/77)	
Publication No. 96729000	

REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4455 Eastgate Mall
La Jolla, California 92037
or use Comment Sheet in the back of this manual.

MANUAL TO EQUIPMENT LEVEL CORRELATION SHEET

This manual reflects the equipment configurations listed below.

EXPLANATION: Locate the equipment type and series number, as shown on the equipment FCO log, in the list below. Immediately to the right of the series number is an FCO number. If that number and all of the numbers underneath it match all of the numbers on the equipment FCO log, then this manual accurately reflects the equipment.

EQUIPMENT TYPE	SERIES	WITH FCOs	COMMENTS
DT195-A	01		
FC402-A	01		
DT120-A	01		

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Cover	--								
Title Page	--								
ii	A								
iii/iv	A								
v/vi	A								
vii/viii	A								
ix	A								
x	A								
1-1	A								
2-1	A								
3-1	A								
4-1 thru 4-48	A								
5-1	A								
5-2	A								
6-1	A								
A-1 thru A-16	A								
B-1 thru B-3	A								
Comment Sheet	A								
Cover	--								

PREFACE

This manual is to be used in conjunction with the CDC® Basic Micro-Programmable Processor Hardware Maintenance Manual.

It contains the theory of operation, diagrams, and maintenance information for the DT195-A (FC402-A and DT120-A) Breakpoint Controller and Breakpoint Panel. Information presented in this manual is intended for use by maintenance personnel in training and in the field.

Logic diagrams are not provided in this manual. The logic diagrams, parts data, and wire list are included in the field print package.

The Basic Micro-Programmable Processor Hardware Maintenance Manual contains a detailed description of the basic processor itself as well as a listing of applicable documents. Other documents that may be of use to the reader are listed below:

Publication	Publication Number
Basic Micro-Programmable Processor Hardware Maintenance Manual	39451400
CW212-A I/O-TTY Controller Hardware Maintenance Manual	96728900
DT195-A Field Print Package	96750800
1700 Enhanced Processor with MOS Memory System Overview Hardware Maintenance Manual	96767800

CONTENTS

FIGURES

TABLES

GENERAL DESCRIPTION

This manual contains the functional and physical descriptions of the breakpoint controller and breakpoint panel.

FUNCTIONAL DESCRIPTION

The breakpoint controller and breakpoint panel, when used with the basic micro-programmable processor, provide a man/machine interface. The breakpoint panel is the device for inputting the control commands and displaying the responses. The breakpoint controller receives the control command, executes it in conjunction with the processor, and returns a response to the breakpoint panel. The breakpoint controller also has the capability of interfacing with the remote devices shown in figure 1-1.

PHYSICAL DESCRIPTION

The breakpoint controller consists of one 11- by 14-inch printed circuit board that plugs into a slot of the processor chassis. The breakpoint panel consists of a 16- by 4-1/2-inch printed circuit board that is mounted directly above the processor chassis. It connects to the panel interface module through a flexible ribbon cable.

ELECTRICAL DESCRIPTION

The breakpoint controller requires +5 V dc and ± 12 V dc, which are derived from the main central processing unit (CPU) power supply via backplane pins. The breakpoint panel requires only +5 V dc, which is provided by the breakpoint controller via the flexible cable.

There are three types of logic level in the breakpoint controller:

- The breakpoint controller operation is performed using standard TTL logic level signals:

Logical zero (low)	+0.4 V dc or less
Logical one (high)	+2.4 to 5.25 V dc

- The RS232-C logical level is required for interfacing with any RS232-C-compatible device:

Logical zero (low)	-3 V dc to -12 V dc
Logical one (high)	+3 V dc to +12 V dc

- The 20 mA current loop is required for interfacing with the teletypewriter.

ENVIRONMENTAL CONDITIONS

Operational:

Temperature – 40°F to 120°F (4.4°C to 48.4°C) (maximum thermal shock 0.2°F/min.)

Relative Humidity - 10 to 90 percent.

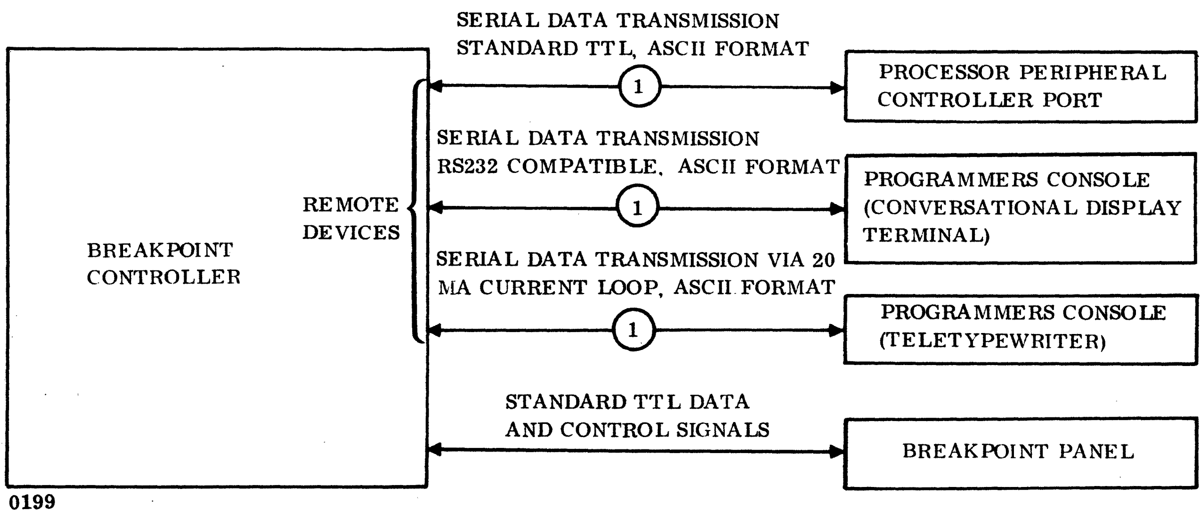
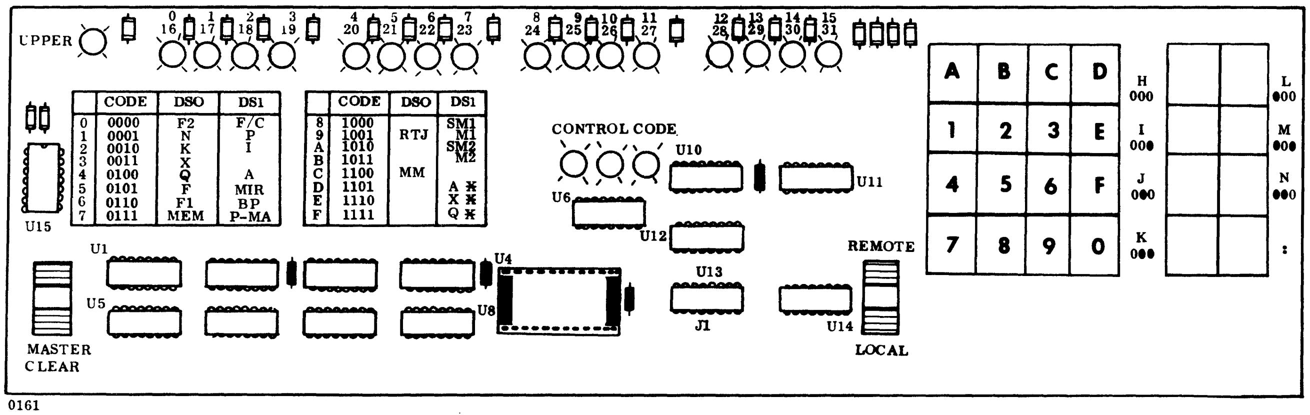

SERIAL DATA TRANSMISSION
STANDARD TTL, ASCII FORMAT
1
PROCESSOR PERIPHERAL CONTROLLER PORT
SERIAL DATA TRANSMISSION
RS232 COMPATIBLE, ASCII FORMAT
1
PROGRAMMERS CONSOLE (CONVERSATIONAL DISPLAY TERMINAL)
REMOTE DEVICES
BREAKPOINT CONTROLLER
SERIAL DATA TRANSMISSION VIA 20 MA CURRENT LOOP, ASCII FORMAT
1
PROGRAMMERS CONSOLE (TELETYPEWRITER)
STANDARD TTL DATA AND CONTROL SIGNALS
BREAKPOINT PANEL
0199

Figure 1-1. External Interface Capability of Breakpoint Controller

OPERATION AND PROGRAMMING

Once installed, the breakpoint controller requires no operating or programming except selecting the desired baud rate via a switch on the I/O-TTY controller. The breakpoint controller control commands are described in detail in section 4, Theory of Operation.

The breakpoint panel switches and display light-emitting diodes (LEDs) are shown in figure 2-1 and are described in table 2-1. The operation of switches is self-explanatory. The control command is entered via the switches in the same order as it is written. The response is displayed by the LEDs.



	CODE	DSO	DS1
0	0000	F2	F/C
1	0001	N	P
2	0010	K	I
3	0011	X	
4	0100	Q	A
5	0101	F	MIR
6	0110	F1	BP
7	0111	MEM	P-MA

	CODE	DSO	DS1
8	1000		SM1
9	1001	RTJ	M1
A	1010		SM2
B	1011		M2
C	1100	MM	
D	1101		A ✱
E	1110		X ✱
F	1111		Q ✱

Figure 2-1. Breakpoint Panel Controls and Indicators

TABLE 2-1. BREAKPOINT PANEL SWITCHES AND INDICATORS (LED)

Switch/Indicator	Type of Switch/Indicator	Function
Data display	16 indicators (LEDs)	Display current binary data as determined by specific control characters
UPPER	1 indicator (LED)	Indicates whether the display is upper or lower 16 bits; upper (0 through 15) when lit
CONTROL CODE	3 indicators (LEDs)	Indicate last control character entered as follows: 0 = H 1 = I 2 = J 3 = K 4 = L 5 = M (Undefined) 6 = N (Undefined) 7 = Error
MASTER CLEAR	Momentary switch	Master clear to CPU, memory,and peripheral controllers
REMOTE/LOCAL	Two-position switch	Enables panel or remote programmers console as follows: REMOTE position = Programmers console is enabled LOCAL position = Panel is enabled
Data entry	16 momentary pushbuttons	For entry of hexadecimal data (0 through F)
Control character	8 momentary pushbuttons	For entry of control character (H through N)
Function control definition table		Operator assistance information such as function control register definition and control character definition

INSTALLATION AND CHECKOUT

INSTALLATION

The breakpoint controller is normally installed in slot U of the processor chassis. Refer to the card-slot assignment in the 1700 Enhanced Processor with MOS Memory System Overview Hardware Maintenance Manual. The breakpoint panel is mounted directly above the processor using the four screws that are provided. Install a flexible ribbon cable between the breakpoint controller and the breakpoint panel.

NOTE

Be sure that the arrows on the male connectors at both ends of the cable are pointed to pin number 1 of the female connectors on the boards.

CHECKOUT

The breakpoint controller and the breakpoint panel can be checked out using the following general procedure. Refer to Breakpoint Controller Functional Operations in section 4 for details about the control commands.

1. Select a parameter in either display 1 or display 0 (i.e., the P register, micro memory, macro memory, or function control register).
2. Enter the selected parameter with the desired data.
3. Display the selected parameter.
4. Compare the displayed data with the entered data for correctness.
5. Repeat the above procedure for all parameters of displays 1 and 0. Also verify that H and I control commands are executed correctly.

Refer to section 3 of the Basic Micro-Programmable Processor Hardware Maintenance Manual for some typical panel interface operation procedures, such as enter/display a register, read/write micro memory, and read/write macro memory.

The success of the above operations indicates that the breakpoint controller and breakpoint panel are fully operational in performing the basic system communications.

THEORY OF OPERATION

This section presents a functional description of the breakpoint controller module and the breakpoint panel. The breakpoint controller is completely described first and the breakpoint panel is described later. The theory of operation for the breakpoint controller consists of the controller functional operations and implementation of these operations using both hardware and controlware. Some typical operations are described at the end to complete the combined operation of hardware and controlware.

BREAKPOINT CONTROLLER FUNCTIONAL OPERATION

The breakpoint controller provides for all common computer control panel functions, such as enter/display registers, read/write macro memory, read/write micro memory, and breakpoint, as well as start, stop, and master clear functions via the breakpoint panel or the programmers console. The programmers console can be either a teletypewriter (20 mA current loop), a conversational display terminal (RS232-compatible), or any device with TTL logic level that has full duplex serial data transmission and ASCII format.

FUNCTION CONTROL REGISTER

A basic part of the breakpoint controller operation is the function control register (FCR). It is a 32-bit register that has access to the CPU similar to the way switches on a conventional panel have access to the CPU (see figure 4-1). FCR bit assignments are listed in tables 4-1 and 4-2.

The 32 bits of the FCR can be grouped into eight hexadecimal digits (digit 0 is the highest order) as follows:

- Display – Digits 0 to 1
- Machine Modes – Digits 2 through 5
- Machine Status – Digits 6 and 7

The display digits determine which individual register of two groups of registers (shown in table 4-2) can be displayed and/or modified. The machine modes, digits 2 through 5, are used to set such conditions as step/run mode, selective stop, selective skip, etc., in the same manner as switches on a conventional control panel. The two least significant digits (6 and 7) of the FCR indicate the status of the processor, such as overflow, parity error, protect fault, etc.

TABLE 4-1. FUNCTION CONTROL REGISTER

Bit		Digit	Bit definition
00	00	(LSB)	Overflow
01	01	7	Protected instruction
02	02		Protect fault
03	03		Parity error
04	04		Interrupt system active
05	05	6	Auto-restart enabled
06	06		Micro running
07	07		Macro running
08	08		
09	09	5	
10	0A		Enable auto-display
11	0B		Enable console echo
12	0C		Enable micro memory write
13	0D	4	Multilevel indirect address mode
14	0E		
15	0F		Suppress console transmit
16	10		0 0 Breakpoint off; 0 1 Instruction reference breakpoint
17	11		1 0 Storage operand breakpoint; 1 1 All references breakpoint
18	12	3	Breakpoint instruction (breakpoint stop if clear)
19	13		Micro breakpoint, step, go, stop (macro if clear)
20	14		Step
21	15	2	Selective stop
22	16		Selective skip
23	17		Protect switch
24	18		
25	19	1	
26	1A		Display 1
27	1B		
28	1C		
29	1D		
30	1E	0	Display 0
31	1F	(MSB)	

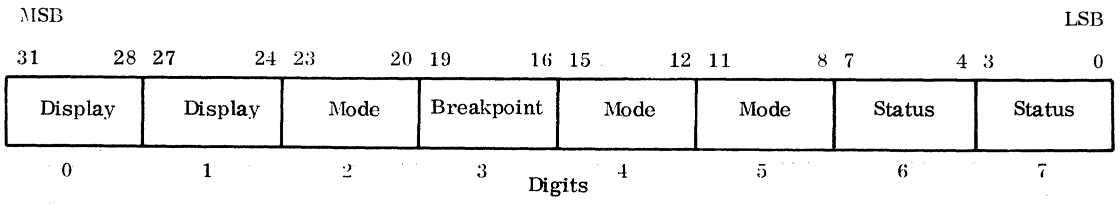


Figure 4-1. Function Control Register

TABLE 4-2. DISPLAY CODE DEFINITIONS

Code	Select Code	Display 1 (K Function)	Select Code	Display 0 (L Function)
0 0000	J10:	FCR	J00:	F2 (Addressed by N register)
1 0001	J11:	P†	J01:	N (MSBs)††
2 0010	J12:	I†	J02:	K (LSBs)††
3 0011			J03:	X
4 0100	J14:	A†	J04:	Q
5 0101	J15:	MIR	J05:	F
6 0110	J16:	BP-P/MA	J06:	F1 { Addressed by K register / Enabled by SM111
7 0111	J17:	BP-P/MA (Display only)	J07:	MEM
8 1000	J18:	SM1		
9 1001	J19:	M1	J09:	~~RTJ~~
A 1010	J1A:	SM2		
B 1011	J1B:	M2		
C 1100			J0C:	MM
D 1101	J1D:	A*		
E 1110	J1E:	X*		
F 1111	J1F:	Q*		

†Used to address macro memory. The P and A registers are automatically incremented after each memory reference. The I register is not incremented after the memory reference.

††The combined contents of these two registers are used to address micro memory. The K register is automatically incremented after each memory reference. The N register does not automatically increment.

The following are some restrictions for the use of FCR bits.

- FCR bits FC10 and FC11 (enable console echo and enable auto-display) are mutually exclusive; that is, the operator may select one or the other but not both simultaneously.

- Selecting BP-P/MA code 0 1 1 1 from table 4-2 results in both the breakpoint (BP) register and the page/memory address register (BP-P/MA) being displayed. BP is the leftmost 16 bits, and P/MA is the rightmost 16 bits. BP can only be modified if BP-P/MA code 0 1 1 0 is selected. P/MA cannot be modified in either case.

- Selecting the N or K register (table 4-2) results in both N and K being displayed. N is the leftmost eight bits, and K is the rightmost eight bits. However, the N register can only be modified when N is selected. Similarly, the K register can only be modified when K is selected.

AUTO-DISPLAY

When auto-display is enabled (FCR bit FC10 is set), the register selected by the control code and display codes is output to the operator's interface and continuously updated as long as the interface is a display terminal and not a teletypewriter. With auto display enabled, depressing a terminator (:, G, or @) with no control character preceding it causes a go signal. This method can be used for stepping through a micro or macro program.

BREAKPOINT

There are two types of breakpoint: micro and macro. If FCR bit FC19 is set, micro breakpoint is selected. If bit FC19 is clear, macro breakpoint is selected.

Macro Breakpoint

Bits FC16 and FC17 of the FCR are used to select three types of macro breakpoint.

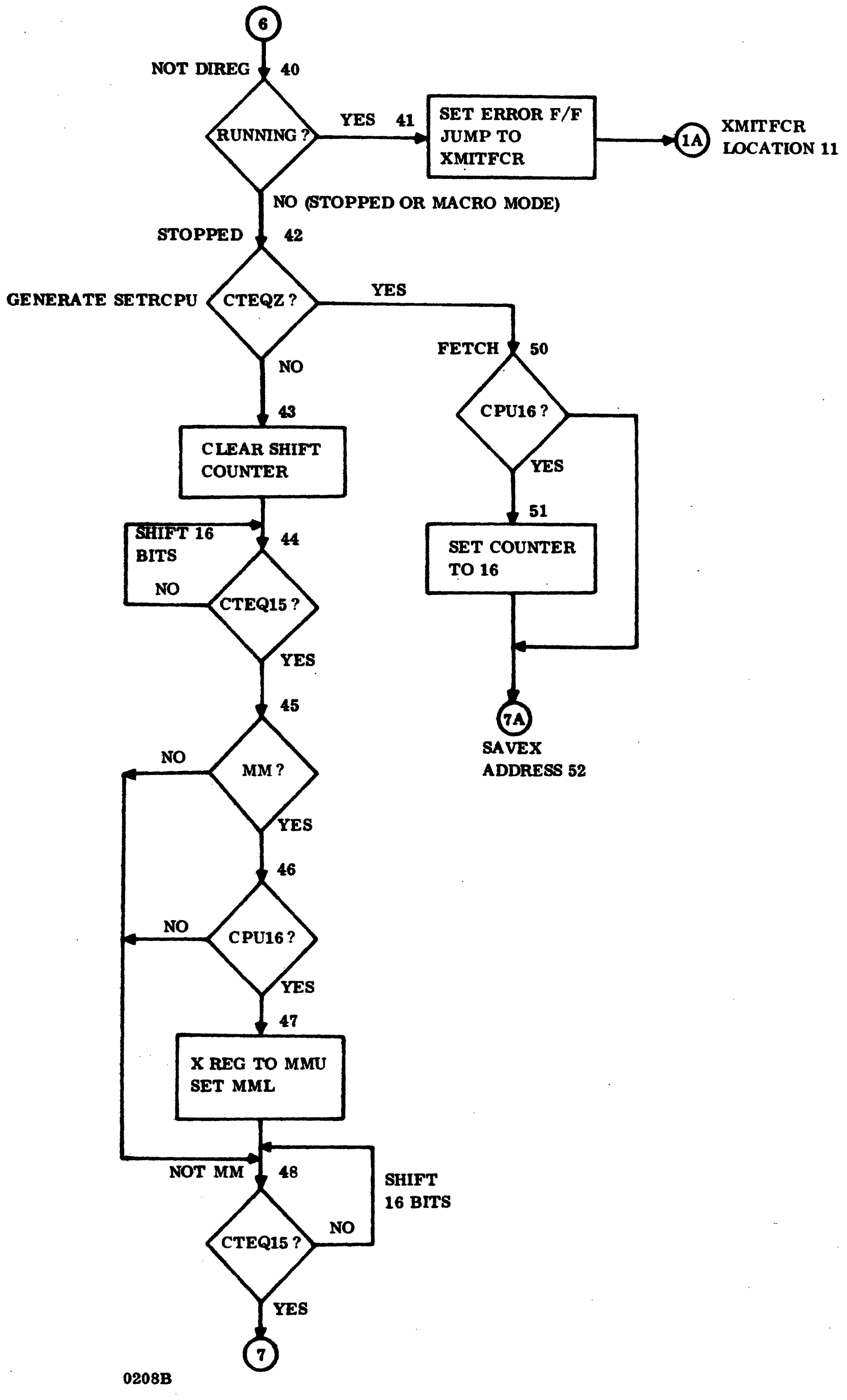
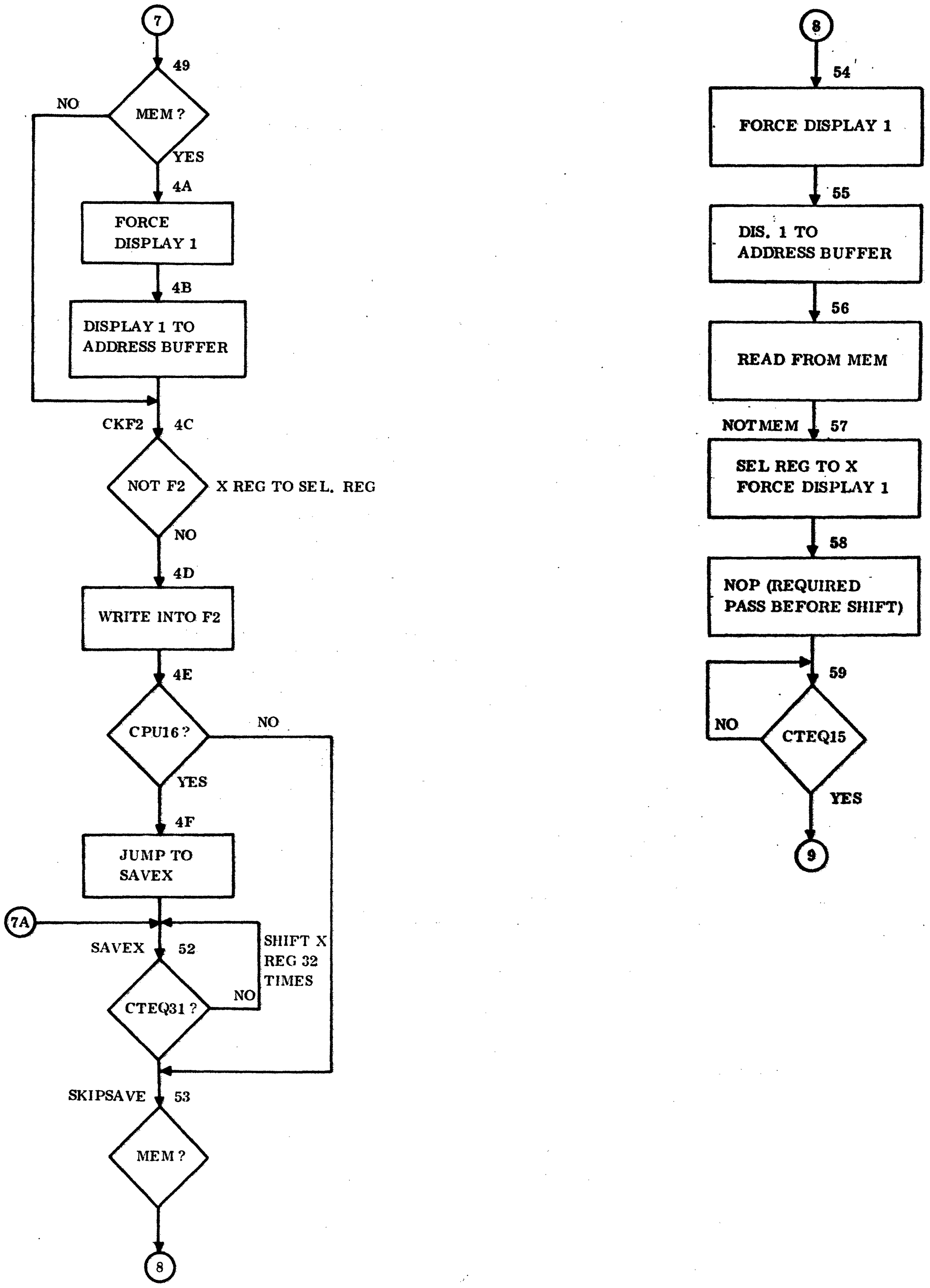
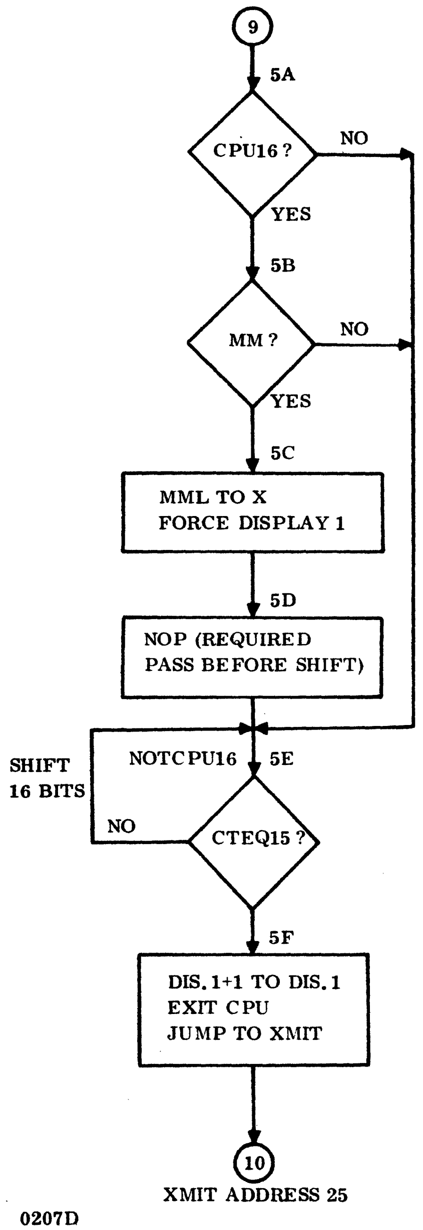


Figure 4-13. Change/Fetch Operation Flow Chart (Sheet 3 of 5)



207C

Figure 4-13. Change/Fetch Operation Flow Chart (Sheet 4 of 5)



0207D

Figure 4-13. Change/Fetch Operation Flow Chart (Sheet 5 of 5)

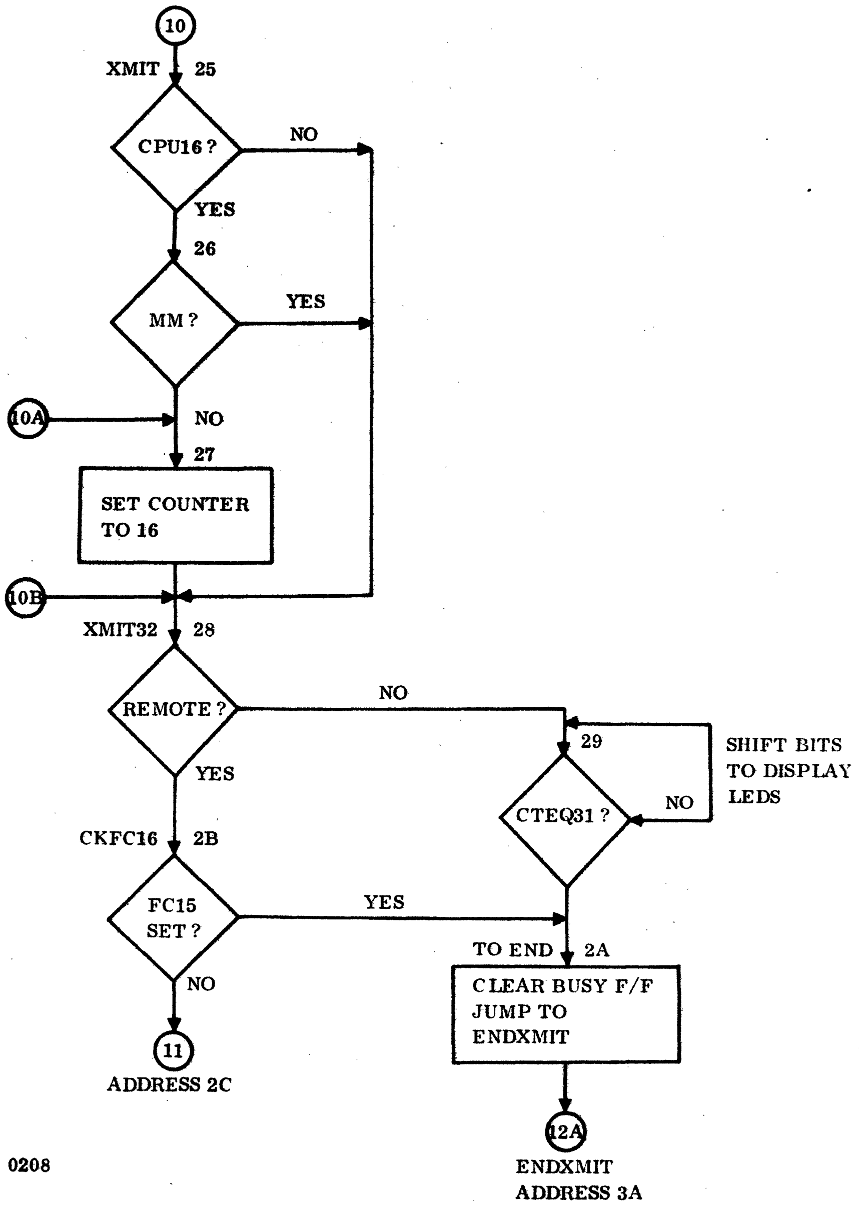
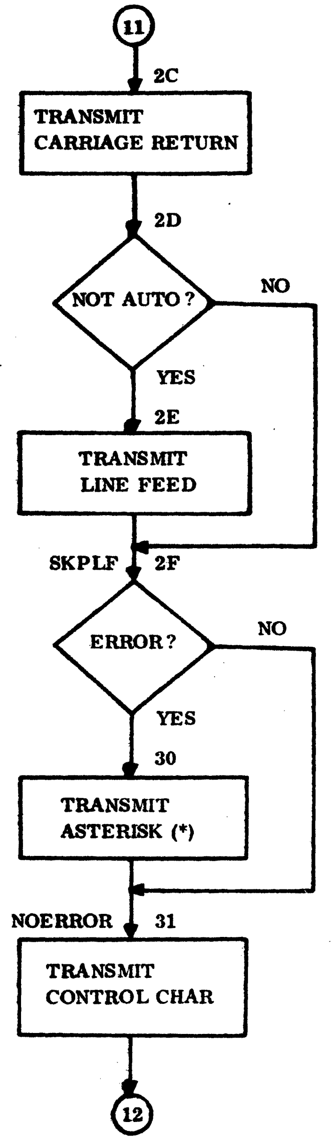


Figure 4-14. Transmit Operation Flow Chart (Sheet 1 of 3)



0208A

Figure 4-14. Transmit Operation Flow Chart (Sheet 2 of 3)

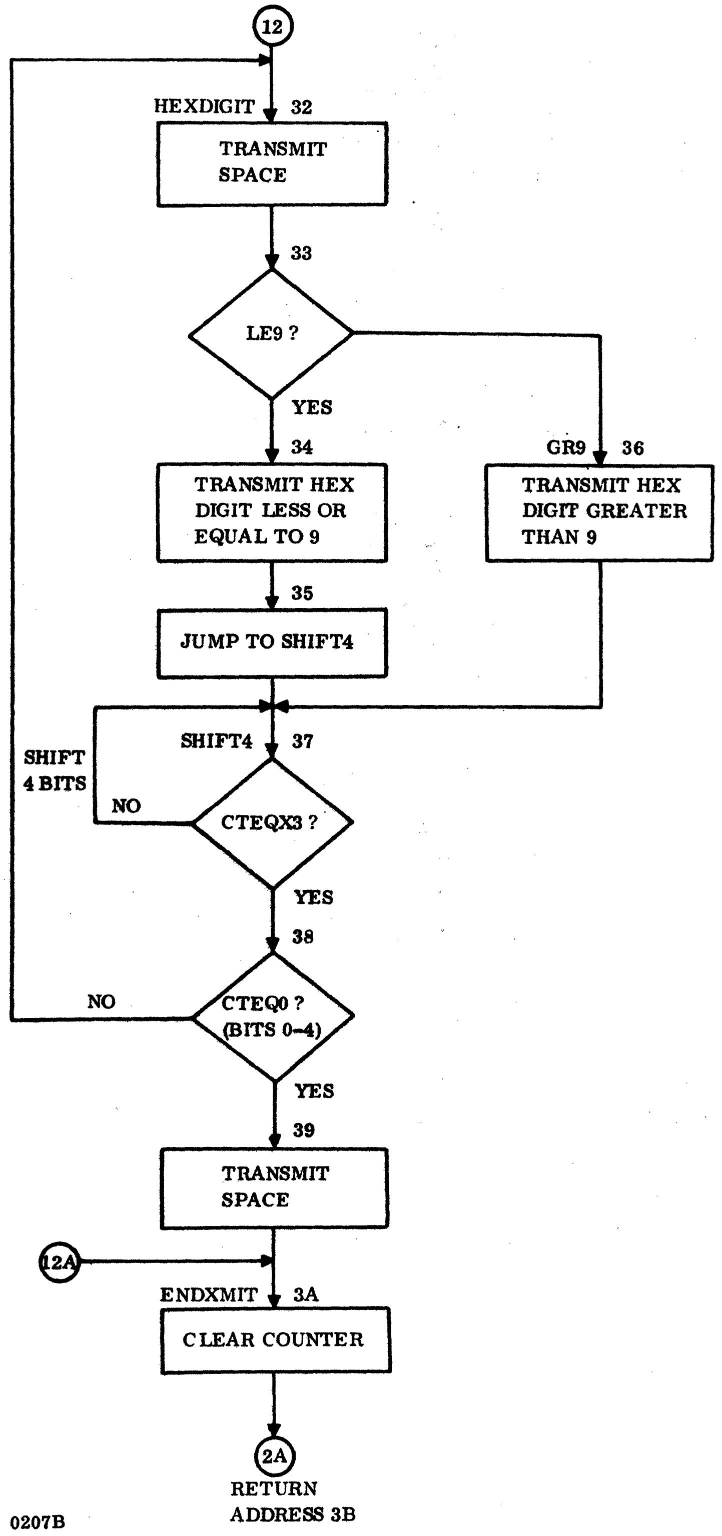
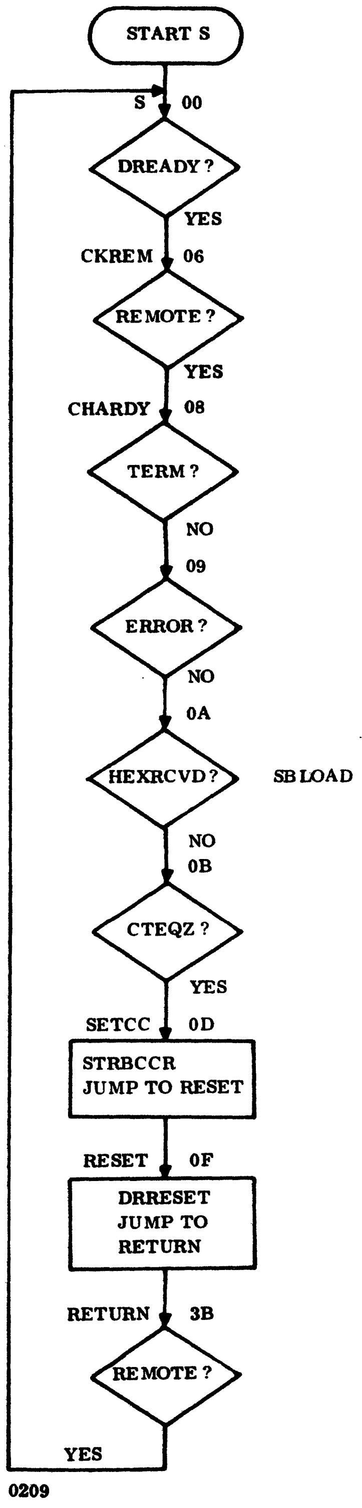


Figure 4-14. Transmit Operation Flow Chart (Sheet 3 of 3)



REMARKS

Data ready (DREADY) condition is true when one complete character has been received at the UART or any panel key is depressed.

The remote/local switch on the panel is in the REMOTE position.

The J control character is not a terminator.

The error flip-flop is not set (no transmit error or no illegal code).

The J control character is not a hexadecimal digit. Generate SBLOAD to load the shift buffer register.

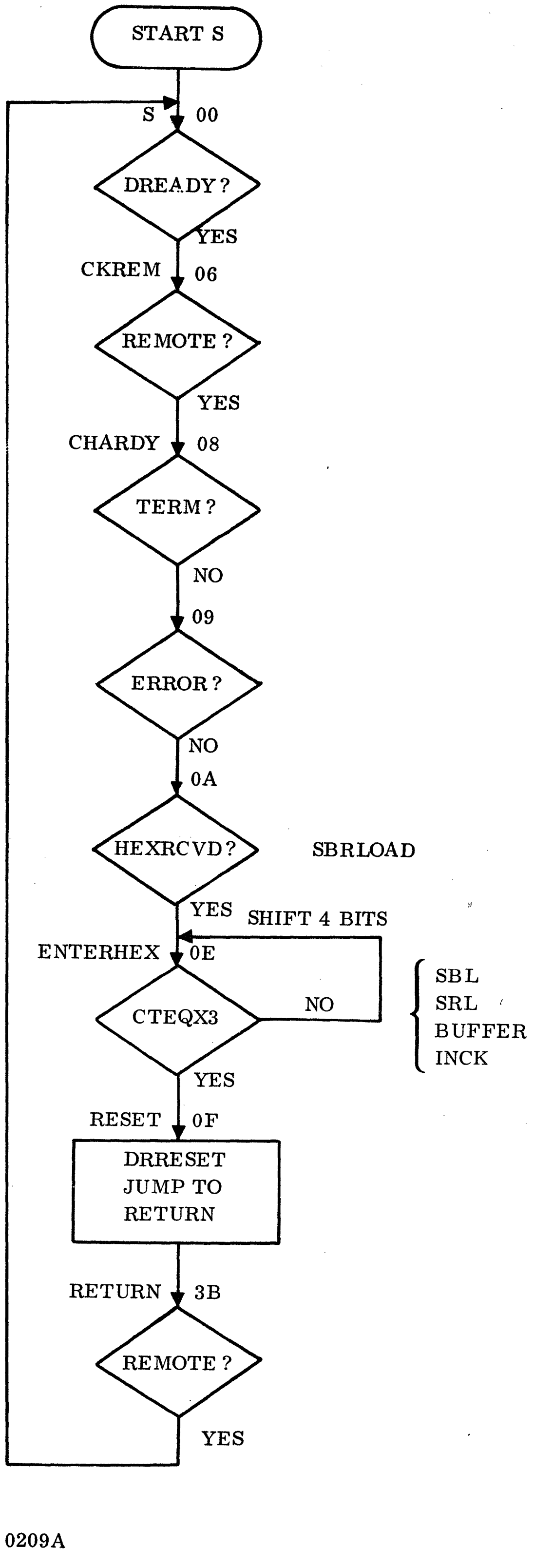
Bits 0 through 5 of the shift counter must equal 0. This step assures that no other character has been entered prior to control character J.

Generate STRBCCR to load the control character register with J code. Jump to the RESET control instruction.

Generate DRRESET to reset the data ready line of the UART. Jump to RETURN.

The REMOTE condition is true; go back to S.

Figure 4-15. Input Operation Flow Chart (Enter J) (Sheet 1 of 3)



REMARKS

1_{16} is the hexadecimal digit. SBLOAD = Load the shift buffer.

Transfer the contents of the shift buffer to the shift register by:

- SBL = Shift shift buffer to left one bit.
- SRL = Shift shift register to left one bit.
- BUFFER = Select output of shift buffer to be left shift serial input to shift register.
- INCK = Enable shift counter.

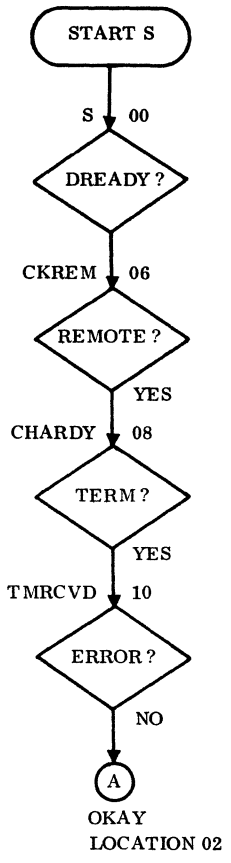
Repeat the above control instruction four times.

Reset the data ready line from the UART.

0209A

Figure 4-15. Input Operation Flow Chart (Enter J) (Sheet 2 of 3)

REMARKS

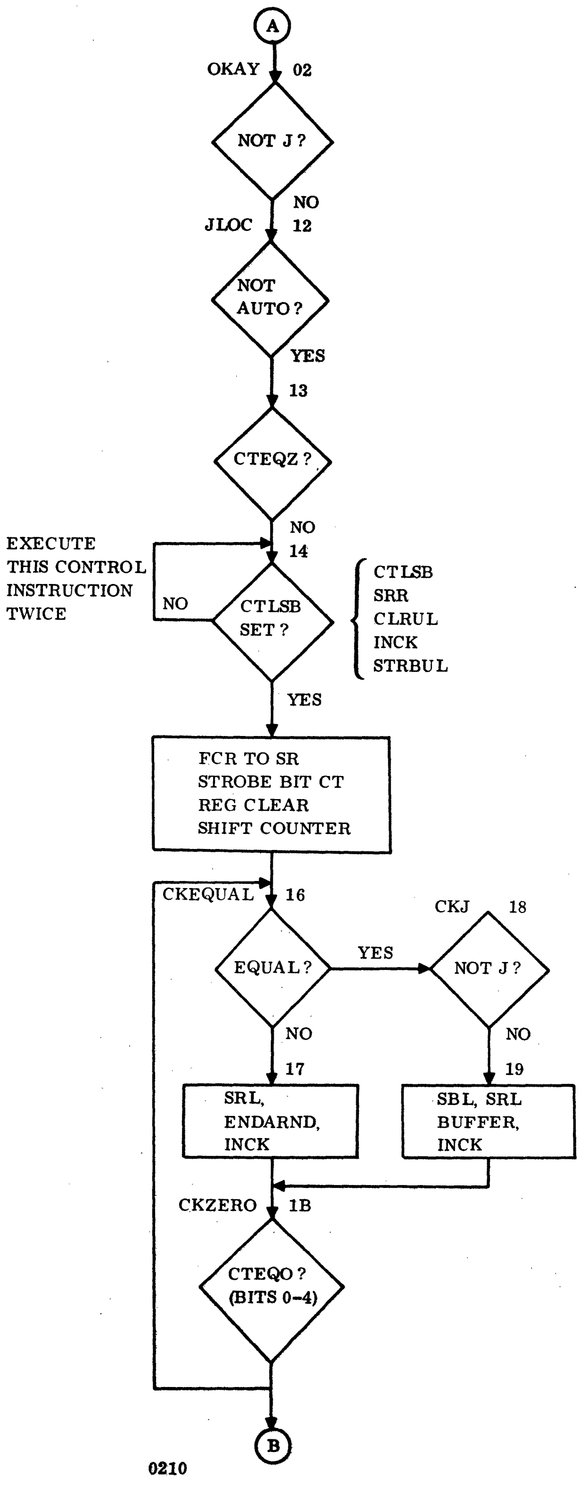


The colon (:) is a terminator.

DRRESET = Reset the data ready line of the UART. Jump to the OKAY control instruction, which is the beginning of the change/fetch operation.

0209B

Figure 4-15. Input Operation Flow Chart (Enter J) (Sheet 3 of 3)



REMARKS

J control command

Bits 0 through 5 of the shift counter equal 001000. The CTEQZ condition is not true.

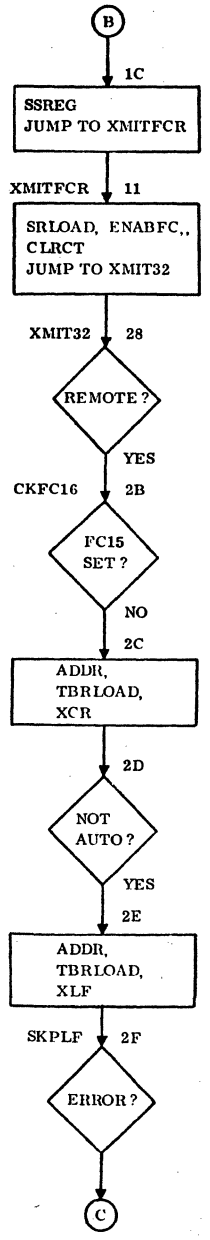
This control instruction is executed twice to shift the shift register to the right two places.

	Initially		After two instruction executions
LSB of SR =	00010001	to	01000100
Shift counter =	001000	to	001010

Bit count register = 00100. Load the shift register with the contents of FCR, 12345678_{16}. Clear the shift counter (shift counter = 000000).

Note that during this J operation, the lower two bits of the bit counter register and the lower two bits of the shift counter are not compared and logically forced to be equal. (Refer to the hardware description of the bit count register.) Initially, the shift register is shifted left four bits with end-around (shift register = 23456781_{16}). The EQUAL condition is now true. (Bit counter register = shift counter = 001xx). The contents of the shift buffer are transferred to the shift register by shifting left four times (the EQUAL condition is true four times). The shift register now contains 3456781_{16}. The shift register continues to be shifted left with end-around until the CTEQ0 condition is true (bits 0 through 4 of the shift counter = 00000). The shift register now contains 11345678_{16}.

Figure 4-16. Change/Fetch Operation Flow Chart (J11: Command)



0211

REMARKS

Generates SSREG, which in turn generates STROBE F/C signal to load the FCR with the contents of shift register.

FCR to SR
Clear shift counter
Jump to MIT 32

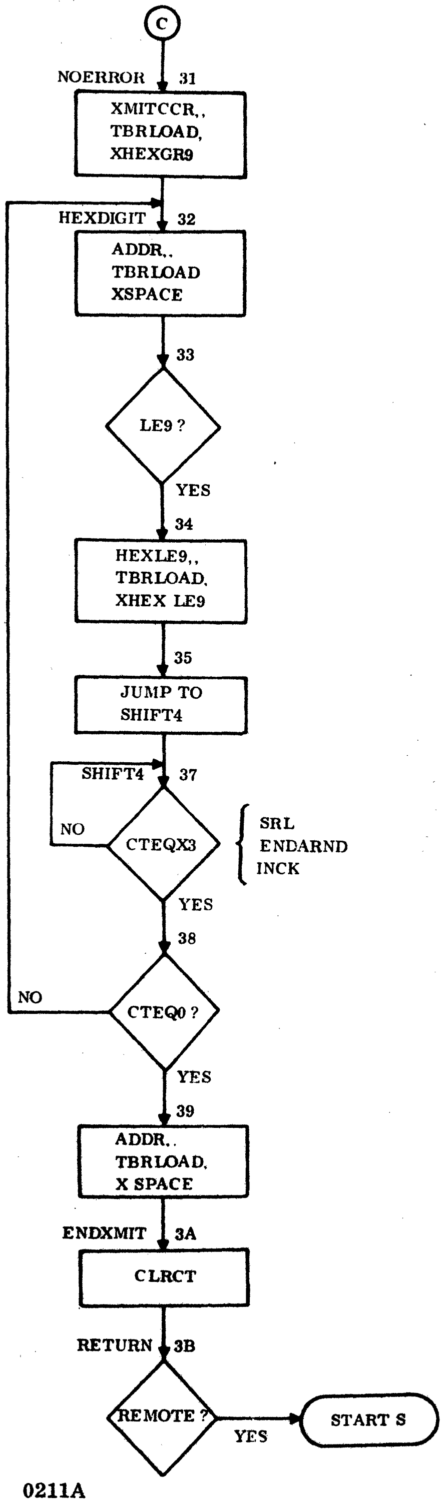
Function control register bit FC15 is previously assumed not set (not suppressing console transmit).

Transmit carriage return ASCII code by loading the transmit buffer register of the UART with address field P7 equal to 0001101.

Assume not auto display mode.

Transmit line feed ASCII code by loading the transmit buffer register of the UART with address field P7 equal to 0001010.

Figure 4-17. Transmit Operation Flow Chart (J11: Command) (Sheet 1 of 2)



0211A

REMARKS

Transmit control character J in ASCII code.

Transmit a space in ASCII code.

Check if the hexadecimal digit is less than 9, and clear the busy flip-flop.

Transmit any hexadecimal digit less than 9 in ASCII code.

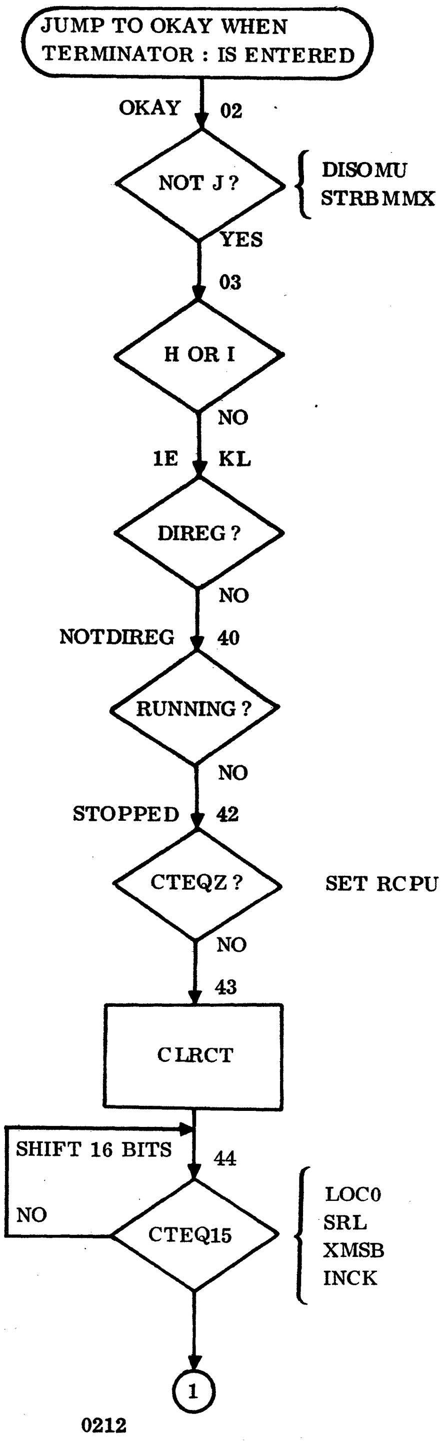
Shift the shift register to the left, end-around, four bits. This step selects the next hexadecimal digit for transmission.

Check if bits 0 through 4 of the shift counter equal 0. If not, repeat the transmit hexadecimal digit less than 9 subroutine (all hexadecimal digits of the FCR in this example are less than 9).

Transmit a space in ASCII code.

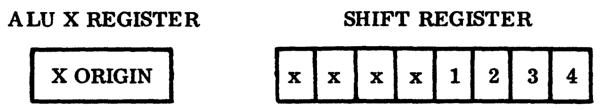
Clear the counter.

Figure 4-17. Transmit Operation Flow Chart (J11: Command) (Sheet 2 of 2)



REMARKS

After the input operation (K1234: command has been entered) the ALU X register contains its original value and the shift register contains 1234_{16}.



P is not the direct register.

Assume that the processor is in micro mode and not running. Otherwise an error occurs.

SETRCPU from controlware sets SM214 to 1 to take over control of the CPU. SETRCPU also generates the RCPU signal, which conditions the panel interface hardware to processor timing.

Clear the shift counter.

LOC0 micro instruction:

0	1	0	1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	Y
		B					DISP 1			X			X							NOP											

This micro instruction shifts the X register left one bit with SR31 as left shift input data. The shift register is also shifted left with XMSB as the left shift input data. After 16 shifts:

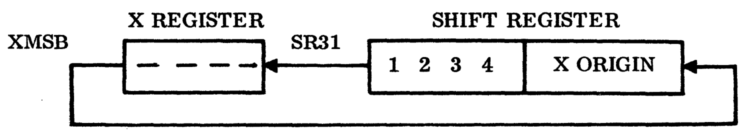
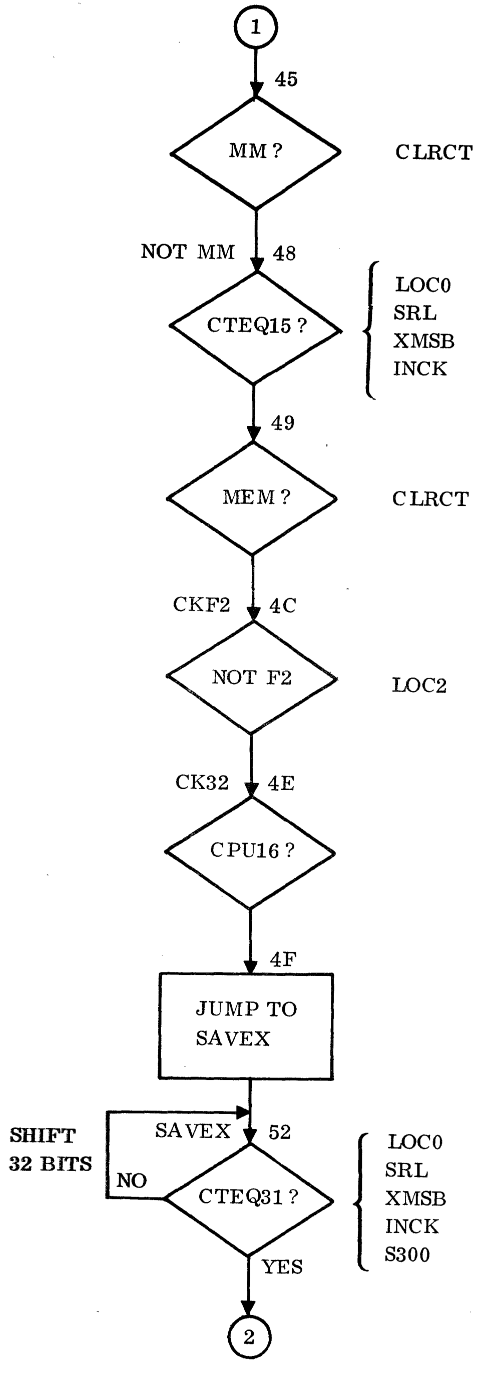


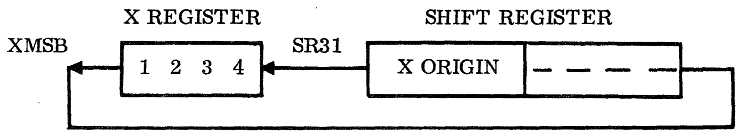
Figure 4-18. Enter P with 1234 command (K1234:), Change/Fetch Operation (Sheet 1 of 3)



REMARKS

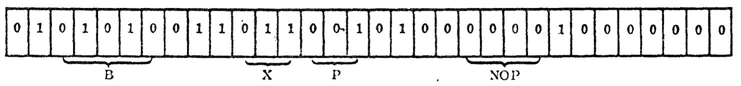
Clear the shift counter. Jump to NOTMM.

Shift the X register and shift register 16 bits.



Clear the shift counter. Jump to CKF2.

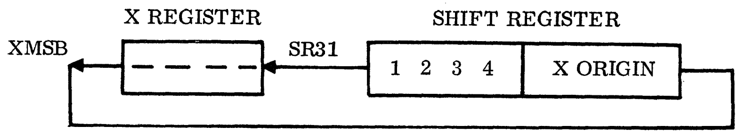
The LOC2 micro instruction transfers the contents of the X register to the selected register, P:



The P register now contains 1234_{16}.

Do not jump if there is a 16-bit CPU.

The X register and shift register are both shifted left 32 bits. Since X is not the selected register, its original contents must be saved. The P6 field equal to 0001 is decoded into the XENAB signal, which selects SR31 (save X since it is not the selected register) as the least significant bit of micro instruction LOC0.



0212A

Figure 4-18. Enter P with 1234 command (K1234:), Change/Fetch Operation (Sheet 2 of 3)

Bit 17	Bit 16	
0	0	Breakpoint not selected
0	1	Instruction reference breakpoint
1	0	Store operand breakpoint
1	1	All references breakpoint

A macro breakpoint stop occurs when the breakpoint register is equal to the macro-memory address and the select conditions are met. If FCR bit FC18 is set, an interrupt rather than a stop occurs when the breakpoint conditions are met.

Micro Breakpoint

FCR bits FC16 and FC17 are used to select two types of micro breakpoint.

Bit 17	Bit 16	
0	0	Breakpoint not selected
0	1	Upper/lower micro instruction breakpoint
1	0	Micro-word breakpoint
1	1	Micro-word breakpoint

The upper/lower micro instruction breakpoint (FC17-FC16 = 01) requires that the micro memory address (P/MA) and upper/lower micro instruction selections are equal to the lower 13 bits of the breakpoint register to cause a micro stop. The micro-word breakpoint only requires that the 12 bits of the micro-memory address register (P/MA) are equal to the lower 12 bits of the breakpoint register.

BREAKPOINT CONTROLLER CONTROL COMMANDS

The control characters accepted by the breakpoint controller include: H, I, J, K, L, @, :, G, and ?. Control characters H through L identify the type of data or operation entered or returned. The at symbol (@), the colon (:), and G all perform an entry termination function. The question mark (?) generates a master clear. A normal control command consists of one control character (H, I, J, K, or L); zero, two, four, or eight hexadecimal digits (0 through F); and a terminating entry (:, @ , or G) in that order.

A normal response consists of a control character identifying the data that follows and four or eight hexadecimal digits. If a transmission or operator error occurs on the control command entry, an asterisk (*) precedes the control character and the FCR is unconditionally displayed with the last legal control character. All entries except the ? cause a response, unless FCR bit FC15 (suppress console transmit) is set. The following are control character usage and examples. The colon (:) is used as the terminating entry.

H Control Character

The H control character is used to clear a specific bit in the FCR; for example, H14:.

This would clear FCR bit 14_{16} (starting from the most significant bit (MSB) of FCR bit FC31) and the response would be a display of the updated FCR.

The H control character is also used to generate a stop if H: is entered. This is a micro stop if FCR bit FC19 is set and a macro stop if FCR bit FC19 is clear. The response to this stop command is a display of FCR.

I Control Character

The I control character is identical to H except it sets a specific bit in the FCR. The I control character is also used to generate a go signal if I: is entered. This is a micro go if FCR bit FC19 is set. It is both a micro go and a macro go if FCR bit FC19 is clear.

J Control Character

The J control character is used to replace the contents of the FCR in a digit mode. While it may be used to change the value of any FCR digit, it is generally used to change digits 0 and 1. The value of digits 0 and 1 specifies which parameter is displayed on display requests or entered on enter requests (refer to table 4-2). The J control command always consists of J followed by two hexadecimal digits and a terminator, normally a colon (:). The first hexadecimal digit specifies one of FCR digits 0 through 5 and the second hexadecimal digit specifies the value the digit is to assume, 0 through F. The control command J14: would set FCR digit 1 to 4_{16} (select the A register), and the response would be a display of the updated FCR.

The J control character is also used to display alternately the upper and lower 16 bits of a 32-bit register on a 16-bit breakpoint panel. The J: control command causes the other 16 bits to be displayed, and the UPPER indicator on the breakpoint panel is complemented.

K Control Character

The K control character is used to display or enter data into the parameter specified by display 1. The K control command has two formats. The first format, K:, is a request to display the parameter specified by display 1. The second format is an enter data request. It consists of K followed by four to eight hexadecimal digits followed by a terminator. The hexadecimal digits are the data to be entered. For example:

- To display the P register, enter:

J11:	Select P register in display 1 (set FCR digit 1 to 1_{16})
K:	Display P register

- To enter $14FE_{16}$ into the breakpoint register, enter:

J16:	Select breakpoint register (set FCR digit 1 to 6_{16})
K14FE:	Enter $14FE_{16}$ into breakpoint register

L Control Character

The L control character is operationally the same as the K control character, except that it is associated with display 0.

NOTE

When macro memory is displayed or entered, the register selected in display 1 is the macro-memory address. The display 1 selection must be the P, A, or I register. The selected P or A register is automatically incremented by 1 after the display. The I register is not incremented.

When micro memory is displayed or entered, the K register is the least significant eight bits of the address and N register bits N3 through N7 provide the remaining address bits. The K register is incremented by 1 after the display. The N register is not automatically incremented.

Master Clear

A master clear can be generated in one of several ways:

- A power-on master clear
- The master clear switch on the breakpoint panel
- An active low signal from a peripheral controller
- A question mark (?) code input from the programmers console

DEADSTART

The breakpoint controller also provides a deadstart capability that is nothing more than an automated version of keyboard transmission. Deadstart operation provides an automated method of loading data into micro or macro memory and all registers. During deadstart operation, the controller executes the control commands as they are continuously input from a deadstart device. The deadstart program must be written in terms of control commands that select the desired micro or macro memory, set up the initial address, load the data, and automatically generate a go signal at the end, if desired.

BREAKPOINT CONTROLLER AND PROCESSOR COMMUNICATION

During breakpoint panel operations, the processor is required by the controller to perform part of the operation. The breakpoint panel interface operation can be divided into three basic operations.

- Input operation – During this operation, the controller receives the control command from the breakpoint panel or remote devices.
- Change/fetch operation – During this operation, the control command operation is carried out by both the breakpoint controller and the processor (except that when the selected register is either the micro instruction register (MIR), FCR, or BP-P/MA, the processor is not required). The change operation changes the contents of the selected register with new entered data. The fetch operation reads out the contents of the selected register for display.
- Transmit operation - During this operation, the response of the control command is transmitted to the breakpoint panel or remote device.

The input and transmit operations are performed by the breakpoint controller itself. To perform the change/fetch operation, the breakpoint controller must generate the following basic three-step sequence:

1. Request control of the CPU. The breakpoint controller must first take over control of the processing element. The processor must also prepare itself before giving up control to the breakpoint controller.
2. Generate micro instructions to be executed by the processor for the desired change/fetch operation.
3. Exit the processor to return control. Figures 4-2 and 4-3 show the breakpoint controller and processor communication flow diagrams, depending upon the micro or macro mode of operation.

First the breakpoint controller must determine if it needs the CPU. The CPU is normally required except when MIR, FCR, or BP-P/MA is the selected register. If the processor is needed, the controller generates SETRCPU, and, depending on whether FCR bit FC19 is set or clear, the micro mode or macro mode sequence is generated.

MICRO MODE

When the processor is in micro mode, the breakpoint controller does not require any assistance from the emulator (firmware), but the processor must initially be stopped before the panel interface can take over control of the processor. To take control of the processor, the controller first generates the SETSM214 signal, which causes the processor to:

- Set the status mode register bit SM214. The SM214 pre-enables the breakpoint controller to MIR.
- Clear the micro instruction register (MIR).
- Cause the ENMM signal in the control 1 module to go low, which disables the micro memory and enables the breakpoint controller micro instructions to the micro memory three-state bus.
- Disable the strobe macro-memory address buffer register by causing GATEAB in the control 1 module to be high except for the read/write macro-memory operation.
- Save the micro-memory address by causing DECMAC to be high.
- Save test bit TB by disabling the GATET clock in the control 2 module.

The breakpoint controller now generates the CONSTART signal, which causes the processor to run in micro mode (allowing the main clock generator to run). The no operation (NOP) micro instruction, MIR = 0, is executed. The controller generates a micro instruction sequence with the M field equal to 01, which is executed by the processor to change/fetch the selected register. Prior to execution of the last breakpoint controller micro instruction, the controller generates CLRSM214, which clears status mode bit SM214 to pre-enable the micro memory to MIR. The controller now generates the last micro instruction with the M field equal to 00 and the T field equal to 000 to exit to the CPU. The M field equal to 00 causes the ENMM signal to go high, which enables the micro memory and disables the breakpoint controller to the micro-instruction register. Control is returned to the processor. The processor reads the micro instruction awaiting in MIR prior to the breakpoint controller sequence. The panel interface also stops the processor by generating the CONSTOP signal.

MACRO MODE

When the processor is in macro mode, the emulator is required to prepare the processor before relinquishing control to the breakpoint controller. When the controller determines that it needs the CPU, it generates the RCPU signal which in turn generates the CONSTART signal. The CONSTART signal is required to insure that the micro code is running. Simultaneously, the breakpoint controller

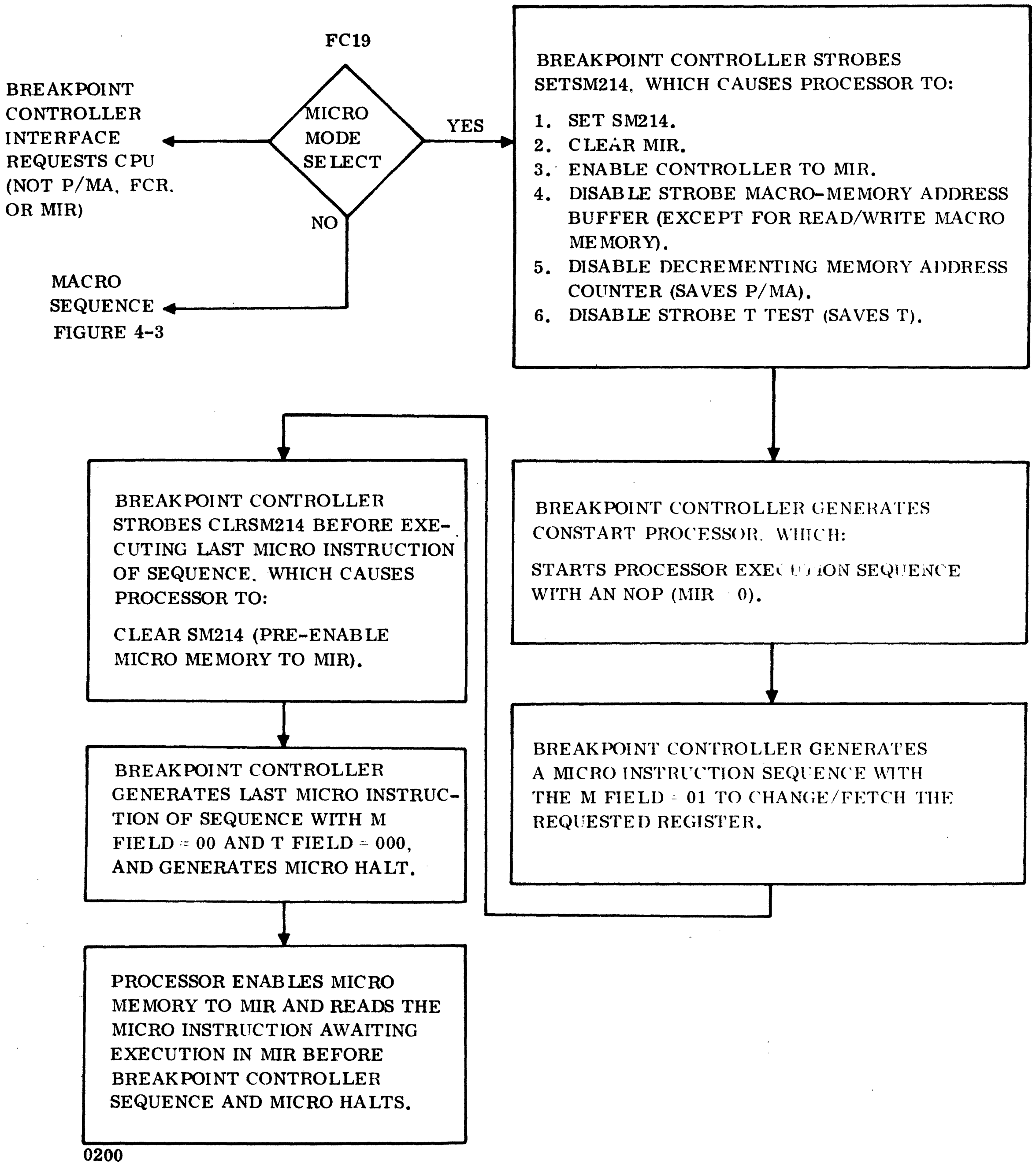
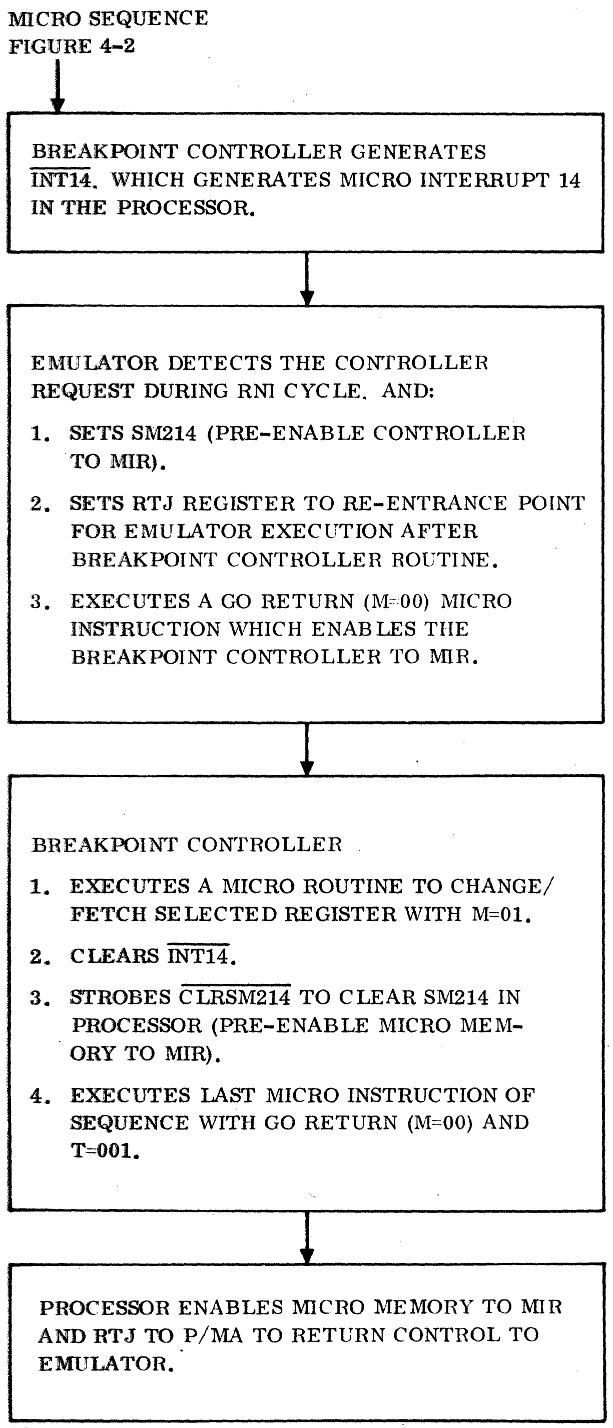

FC19
MICRO MODE SELECT
YES
NO
BREAKPOINT CONTROLLER INTERFACE REQUESTS CPU (NOT P/MA, FCR, OR MIR)
MACRO SEQUENCE FIGURE 4-3
BREAKPOINT CONTROLLER STROBES SETSM214, WHICH CAUSES PROCESSOR TO:
1. SET SM214.
2. CLEAR MIR.
3. ENABLE CONTROLLER TO MIR.
4. DISABLE STROBE MACRO-MEMORY ADDRESS BUFFER (EXCEPT FOR READ/WRITE MACRO MEMORY).
5. DISABLE DECREMENTING MEMORY ADDRESS COUNTER (SAVES P/MA).
6. DISABLE STROBE T TEST (SAVES T).
BREAKPOINT CONTROLLER GENERATES CONSTART PROCESSOR, WHICH:
STARTS PROCESSOR EXECUTION SEQUENCE WITH AN NOP (MIR = 0).
BREAKPOINT CONTROLLER GENERATES A MICRO INSTRUCTION SEQUENCE WITH THE M FIELD = 01 TO CHANGE/FETCH THE REQUESTED REGISTER.
BREAKPOINT CONTROLLER STROBES CLRSM214 BEFORE EXECUTING LAST MICRO INSTRUCTION OF SEQUENCE, WHICH CAUSES PROCESSOR TO:
CLEAR SM214 (PRE-ENABLE MICRO MEMORY TO MIR).
BREAKPOINT CONTROLLER GENERATES LAST MICRO INSTRUCTION OF SEQUENCE WITH M FIELD = 00 AND T FIELD = 000, AND GENERATES MICRO HALT.
PROCESSOR ENABLES MICRO MEMORY TO MIR AND READS THE MICRO INSTRUCTION AWAITING EXECUTION IN MIR BEFORE BREAKPOINT CONTROLLER SEQUENCE AND MICRO HALTS.
0200

Figure 4-2. Micro-Mode Flow Diagram

generates micro interrupt INT14, which is the panel request to CPU interrupt. The emulator performs the following operations once INT14 is recognized by the emulator during the read next instruction (RNI) cycle.

- Sets status mode interrupt bit SM214 to pre-enable the breakpoint controller to MIR.
- Sets return jump (RTJ) register to the re-entrance address for emulator execution after the breakpoint controller sequence.
- Executes a GO return (M field equal to 00) micro instruction which causes the ENMM signal in the control 1 module to go low. This disables the micro memory and enables the breakpoint controller to the micro-memory three-state bus.

The controller now has control of the CPU. The controller generates a micro-instruction sequence with the M field equal to 01 to be executed by the CPU. After the CPU completes the required operation, the breakpoint controller exits the CPU by clearing INT14. The CLRSM214 signal is also generated to clear SM214. This pre-enables the micro memory to the micro-memory bus. The last controller micro instruction is executed with the M field equal to 00 and the T field equal to 001 to exit the CPU. The M field equal to 00 causes the ENMM signal to go high, enabling the micro memory and disabling the breakpoint controller to the


MICRO SEQUENCE
FIGURE 4-2

BREAKPOINT CONTROLLER GENERATES $\overline{\text{INT14}}$. WHICH GENERATES MICRO INTERRUPT 14 IN THE PROCESSOR.

EMULATOR DETECTS THE CONTROLLER REQUEST DURING RNI CYCLE. AND:

1. SETS SM214 (PRE-ENABLE CONTROLLER TO MIR).
2. SETS RTJ REGISTER TO RE-ENTRANCE POINT FOR EMULATOR EXECUTION AFTER BREAKPOINT CONTROLLER ROUTINE.
3. EXECUTES A GO RETURN (M=00) MICRO INSTRUCTION WHICH ENABLES THE BREAKPOINT CONTROLLER TO MIR.

BREAKPOINT CONTROLLER

1. EXECUTES A MICRO ROUTINE TO CHANGE/ FETCH SELECTED REGISTER WITH M=01.
2. CLEARS $\overline{\text{INT14}}$.
3. STROBES $\overline{\text{CLRSM214}}$ TO CLEAR SM214 IN PROCESSOR (PRE-ENABLE MICRO MEMORY TO MIR).
4. EXECUTES LAST MICRO INSTRUCTION OF SEQUENCE WITH GO RETURN (M=00) AND T=001.

PROCESSOR ENABLES MICRO MEMORY TO MIR AND RTJ TO P/MA TO RETURN CONTROL TO EMULATOR.

0201

Figure 4-3. Macro-Mode Flow Diagram

micro-memory three-state bus. The M field equal to 00 also selects the contents of the RTJ register as the address of the next micro instruction pair. Note that the RTJ register contained the re-entrance micro-memory address for the emulator. The T field equal to 001 selects the upper micro instruction at the address specified by the RTJ register. Control is now returned to the emulator and it continues at the point that it stopped prior to the controller sequence.

BREAKPOINT CONTROLLER HARDWARE

The breakpoint controller operations are performed by combining hardware and controlware. The breakpoint controller hardware is divided into two types of functional hardware for clarity: data path hardware and control hardware.

BREAKPOINT CONTROLLER DATA PATH HARDWARE

The breakpoint controller data path consists of the selectors, registers, shift register, universal asynchronous receiver/transmitter (UART), etc., that are interconnected as shown in figure 4-4. The data is manipulated by controlware via various data paths.

Function Control Register

The function control register (FCR) is a basic part of the breakpoint controller operation. It is a 32-bit register that has access to the CPU similar to the way switches on a conventional panel have access to the CPU. Table 4-1 shows the FCR bit assignments. The FCR consists of four HEX-D-type flip-flops, D10, F10, G10, and A11, which store the 24 FCR bits, FC8 through FC31. These FCR bits can be altered externally via control characters; FC00 through FC07 reflect processor status and cannot be changed externally. The FCR is cleared by the master clear signal going low. The FCR and the shift register form a complete feedback loop by having the output of one connected to the input of the other. This allows the FCR contents to be altered via the shift register. The FCR is clocked by the STROBE F/C signal, which is generated by controlware. Refer to the paragraph on Control Instruction Decode below.

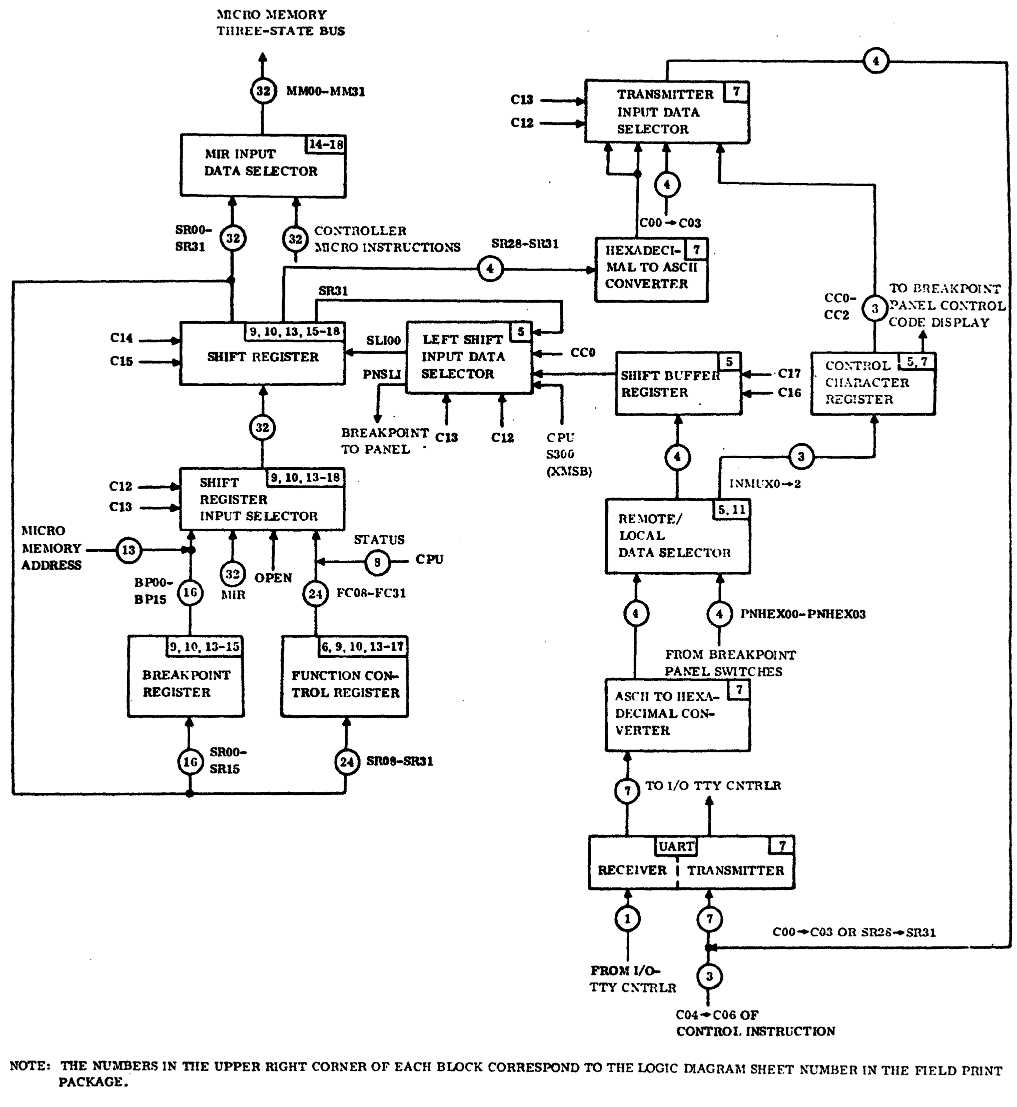
Breakpoint Register

This is a 16-bit register that contains the breakpoint address. When the macro- or micro-memory address is equal to the breakpoint address, the computer is stopped if the breakpoint select conditions are met. The breakpoint register consists of three HEX-D type flip-flops, B10, G11, and K10. The breakpoint register receives its input from shift register output SR00 through SR15, and its output is in turn connected to the input of the shift register. This allows the contents of the breakpoint register to be changed via the shift register. The output of the breakpoint register is also connected to breakpoint comparators to be compared with the macro- or micro-memory address.

Shift Register Input Selector

This selector consists of 4-to-1 multiplexers that select one of the three sources to be parallel-loaded into the shift register for modifying or transmitting. Select signals C12 and C13 are bits 12 and 13 of the 24-bit control instructions. Refer to the paragraphs on Control Instruction Format below. The selector selects the sources as follows:

C13	C12	Selected Source
0	0	Open
0	0	Function control register FC00-FC31
1	0	Micro-instruction register MIR00-MIR31
1	1	Breakpoint register BP00-BP15 and micro-memory address PG0-PG3, MA0-MA7, and BTLW


MICRO MEMORY THREE-STATE BUS
MM00-MM31
MIR INPUT DATA SELECTOR 14-18
SR00-SR31
CONTROLLER MICRO INSTRUCTIONS
SR28-SR31
SR31
C14
C15
SHIFT REGISTER 9, 10, 13, 15-18
SLI00
PNSLI
LEFT SHIFT INPUT DATA SELECTOR 5
CC0
BREAKPOINT TO PANEL
C13
C12
CPU S300 (XMSB)
C12
C13
SHIFT REGISTER INPUT SELECTOR 9, 10, 13-18
MICRO MEMORY ADDRESS
STATUS
CPU
BP00-BP15
MIR
OPEN
FC08-FC31
BREAKPOINT REGISTER 9, 10, 13-15
FUNCTION CONTROL REGISTER 6, 9, 10, 13-17
SR00-SR15
SR08-SR31
TRANSMITTER INPUT DATA SELECTOR 7
C00→C03
HEXADECIMAL TO ASCII CONVERTER 7
SHIFT BUFFER REGISTER 5
C17
C16
CC0-CC2
TO BREAKPOINT PANEL CONTROL CODE DISPLAY
CONTROL CHARACTER REGISTER 5, 7
INMUX0→2
REMOTE/LOCAL DATA SELECTOR 5, 11
PNHEX00-PNHEX03
FROM BREAKPOINT PANEL SWITCHES
ASCII TO HEXADECIMAL CONVERTER 7
TO I/O TTY CNTRLR
UART 7
RECEIVER
TRANSMITTER
C00→C03 OR SR28→SR31
FROM I/O-TTY CNTRLR
C04→C06 OF CONTROL INSTRUCTION

NOTE: THE NUMBERS IN THE UPPER RIGHT CORNER OF EACH BLOCK CORRESPOND TO THE LOGIC DIAGRAM SHEET NUMBER IN THE FIELD PRINT PACKAGE.

0202

Figure 4-4. Panel Interface Data Path Block Diagram

Shift Register

The 32-bit shift register consists of eight universal bidirectional shift registers (G12, A12, B12, H12, D12, E12, J12, and M12), which handle all the data transfers serially in and out of the breakpoint controller module. The following is typical of shift register usage:

- During the input operation, the data entered from the breakpoint panel or remote device is shifted left serially into the shift register for storing via the shift buffer register.

- The data stored in the shift register can be shifted into the arithmetic/logical unit (ALU) X register, which in turn is loaded into any selected parameter of the processor.

- The contents of the ALU X register can be shifted left into the shift register and then shifted to the breakpoint panel via SR31 or to the remote device for display via the UART.

The shift register holds the FCR for modification. The control signals of the shift register are bits C14 and C15 of the panel interface control instructions; therefore, shift register operation is totally a function of controlware. Shift register operation is controlled by C14 and C15 as follows:

C15	C14	Operations
0	0	Do nothing
0	1	Left shift
1	0	Right shift
1	1	Parallel loading

The shift number is determined by controlware and tracked by a shift counter. A description of the shift counter is given in the control hardware section. During the left shift operation, shift left serial input data SL100 is derived from the output of the left shift input selector. The output of shift register SR00 through SR31 is sent to the micro-instruction register via the MIR input data selector and also to the function control and breakpoint registers. This allows modification of these registers by loading the desired data from the shift register. During the display operation, the most significant bit (MSB) SR31 of the shift register is sent out to the breakpoint controller via the left shift input selector if the breakpoint panel is selected. However, if the remote device is selected, the most significant four bits of the shift register, SR28 through SR31, are converted to ASCII code before sending out to the remote device via the universal asynchronous receiver transmitter (UART).

MIR Input Data Selector

The MIR input data selector selects either the output of the shift register, SR00 through SR31, or the panel interface-generated micro instructions to be input to MIR. The selector consists of six 2-to-1 multiplexers, A13, B13, G9, D13, M10, and M11, and four 4-to-1 multiplexers, E11, D11, H9, and H10. The input selection is determined by the RCPU signal. When the RCPU signal is high, it indicates the processing element is under the control of the breakpoint controller; therefore, the controller-generated micro-instruction, MM00 through MM31, is selected. When the RCPU signal is low, SR00 through SR31 is selected to allow MIR to be modified. The MIR input data selector is also a part of controller micro-instruction generation hardware; it is described in detail under Breakpoint Controller Micro Instructions below. The output of the selector is connected to the micro-memory three-state bus so that it can be loaded into the micro-instruction register.

Left Shift Input Data Selector

The left shift input data selector, 4-to-1 multiplexer A6, selects the left shift serial input data SL100 for the breakpoint controller shift register and the left shift serial input data, PNLSI, for the breakpoint panel shift register.

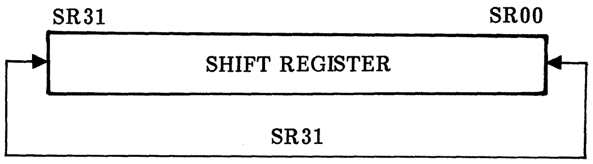
The select signals of the selector are bits C12 and C13 of the control instruction. The input sources are selected as follows:

C13	C12	SL100 for Controller Shift Register	PNLSI for Breakpoint Panel Shift Register
0	0	CC0	Upper flip-flop output (K85)
0	1	SR31	SR31
1	0	S300	SR31
1	1	Output of shift buffer register	Output of shift buffer register

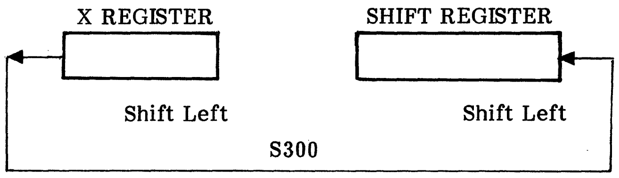
The controlware selects the left shift serial input data for the breakpoint controller shift register according to its desired operation.

- CC0 is the least significant bit (LSB) of the control character code. CC0 is low for the H control character and high for the I control character; therefore, CC0 is selected to clear or set a specific bit in the FCR for the H or I control command, respectively.

- SR31 is the most significant bit of the shift register and is selected when an end-around shift left operation is desired:



- S300 is the most significant bit of central processor ALU selector S3. When X is enabled to the ALU, S300 is the most significant bit of the ALU X register. Therefore, S300 is selected when data transfer from X register to shift register is desired:



- The output of the shift buffer register is selected to transfer data from the shift buffer register to the shift register. This operation is normally required to store data coming in from the breakpoint panel or the remote device via the remote/local data selector during an input operation. It is also used to modify the FCR contents in the digit mode (refer to the paragraph on the control character under Breakpoint Controller Control Commands).

- The left shift serial input data PNLSI for the breakpoint panel is used solely for display purposes.

- The upper flip-flop output at K8-5 is selected to indicate that either the upper or lower 16 bits of data are being displayed via the UPPER indicator on the breakpoint panel. The upper flip-flop output K8-5 is complemented if a J: command is entered, since the J code can be used to display alternately the upper or lower 16-bit data. However, the upper flip-flop output can be forced low by setting bit C12 of the control instruction to high.

- SR31, the most significant bit of the shift register, is selected to transfer the contents of the panel interface shift register to the breakpoint panel shift register to drive the data display LEDs.

- The output of the buffer shift register (left shift serial input data) is required for displaying incoming codes and data during the input operation.

Hexadecimal-to-ASCII Converter

Each hexadecimal digit, which consists of the four most significant bits SR31 through SR28 of the shift register, must be converted into ASCII code before transmitting out to remote devices via the UART. The conversion is accomplished using both hardware and controlware. Table 4-3 shows the ASCII code of the corresponding hexadecimal numbers.

First SR31 through SR28 are decoded to generate the less than or equal to 9_{10} (LE9) condition:

$$LE9 = \overline{(SR29 \cdot SR31) + (SR30 \cdot SR31)}$$

TABLE 4-3. CORRESPONDING HEXADECIMAL NUMBERS AND ASCII CODE

Hexadecimal Number (SR31 - SR28)				ASCII Code						
				6	5	4	3	2	1	0
0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	1	1	0	0	0	1
0	0	1	0	0	1	1	0	0	1	0
0	0	1	1	0	1	1	0	0	1	1
0	1	0	0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	0	1	0	1
0	1	1	0	0	1	1	0	1	1	0
0	1	1	1	0	1	1	0	1	1	1
1	0	0	0	0	1	1	1	0	0	0
1	0	0	1	0	1	1	1	0	0	1
1	0	1	0	1	0	0	0	0	0	1
1	0	1	1	1	0	0	0	0	1	0
1	1	0	0	1	0	0	0	0	1	1
1	1	0	1	1	0	0	0	1	0	0
1	1	1	0	1	0	0	0	1	0	1
1	1	1	1	1	0	0	0	1	1	0
				C06 - C04						

The LE9 condition is checked by controlware. If the LE9 condition is true, the controlware sets bits 4 through 6 of the ASCII code to 011 via three bits, C04 through C06 of the control instruction. Bits 0 through 3 of ASCII remain the same as SR28 through SR31. If the hexadecimal digit is greater than 9, the controlware sets bits 4 through 6 of ASCII code to 100 via bits C04 through C06 of the control instruction. Bits 0 through 3 of ASCII code are converted by hardware from SR28 through SR31 according to the following equations:

ASCII code	Bit 0 = $\overline{\text{SR28}}$
ASCII code	Bit 1 = $\overline{\text{SR28} + \text{SR29}}$
ASCII code	Bit 2 = SR30 • (SR29+SR28)
ASCII code	Bit 3 = Low

Transmitter Input Data Selector

This four-bit selector consists of multiplexers E4 and F2 that select one of four data sources for the UART. The sources are selected by bits C12 and C13 of the control instruction as follows:

C13	C12	Selected Data
0	0	Bits 0 through 4 of the ASCII code correspond to greater than 9_{10} (GR9) hexadecimal digit.
0	1	Bits 0 through 4 of the ASCII code correspond to less than or equal to 9_{10} (LE9) hexadecimal digit.
1	0	Four least significant bits (C00 through C03) of the control instruction
1	1	Control character code CC0 through CC2

Bits C04 through C06 of the control instruction combine with one of the above four data bits to form the complete seven-bit ASCII code applied to the transmitter register data input of the UART. If the control character code is selected, the controlware sets bits C06 to 100 to generate the proper ASCII code for H, I, J, K, and L characters. If select signals C13 and C12 equal 10, the least significant four bits, C00 through C03, of the control instruction are sent out to the programmers console to perform specific operations, such as carriage return, space, line feed, or type an asterisk as indicated by controlware.

Universal Asynchronous Receiver/Transmitter

The detailed block diagram, pin designations, and descriptions of the UART are included in Appendix B. Only a brief description of the UART and how it is used in this breakpoint controller module is given here.

The UART has the capability of simultaneously converting the asynchronous serial binary character to a parallel format (receiver) and parallel characters to serial, asynchronous output (transmitter) with start, parity, and stop bits added or verified.

Since the UART is not required during the time the breakpoint panel is being used (remote/local switch panel is at LOCAL position), the UART is disabled by the LOCAL + MC signal that is derived from the panel switch PNLOCAL signal.

An active high master clear signal also clears the UART. To transmit the character out to the remote devices, the controlware generates the TBRLOAD signal at pin 23 to load the corresponding seven-bit ASCII code into the transmit buffer register. This data is shifted out serially at pin 25 (TRO) via the UART internal transmit register at the rate determined by the clock (PNUARTCLK) at pin 40. Note that the clock frequency is 16 times the desired transmitter shift rate. The transmitter output data is transmitted out to the remote devices via the 20 mA current loop driver for I/O-TTY, an RS232 driver (for the conversational display terminal), and a standard TTL level driver.

The receiver portion of the UART asynchronously receives the serial input data from remote devices at pin 20 (RRI) and converts them into parallel format at the rate determined by the clock, PNUARTCLK. The receiver clock frequency is also 16 times the desired receiver shift rate. The serial data from remote devices is converted to TTL logic level data by a 20 mA current loop receiver, an RS232 receiver, or a standard TTL receiver before being applied to the UART receiver input. Whenever one complete character has been received and assembled, the data ready (RDR) line at pin 19 goes high. The controlware resets the data ready line by generating a DRRESET signal at pin 18 when the received character has been completely processed. Since the data received from remote devices is in ASCII code, it must be converted into hexadecimal for manipulation by the breakpoint controller and the processor.

ASCII-to-Hexadecimal Converter

The conversion from ASCII to hexadecimal is accomplished via the preprogrammed programmable read-only memory (PROM) D1. The conversion is limited to those legal characters shown in table 4-4. The four least significant bits, RR0 through RR3, and the most significant bit RR6 of ASCII code from the UART receiver data outputs are used to address one of 32 code locations. The selected location contains the hexadecimal code corresponding to the ASCII code of the character received.

The characters 0 through F are indicated by bit 7 of the PROM output being a 1. Similarly, the terminators (:), @, and G are indicated by bit 6 of PROM output being 1. These conditions are then used by controlware to determine if a hexadecimal digit, a terminator, or a control character has been received. A code is legal only if bits RR6 through RR4 of the ASCII code are equal to 011 or 100 or if bit 8 of PROM output is a 1.

Remote/Local Data Selector

The 2-to-1 multiplexer C1 selects either the hexadecimal data from the output of the ASCII to hexadecimal converter or the breakpoint panel switches in accordance with the state of the REMOTE signal. The REMOTE signal is high when the remote/local switch of the breakpoint panel is in the REMOTE position or during a deadstart operation (SM204 is low). The output of C1 can be loaded into the shift buffer register and/or to the control character register as determined by controlware. Similarly, the 2-to-1 multiplexer D4 selects the conditions such as the hexadecimal digit receiver (HEXRCVD), terminator code received (TERMCODE), data ready (DRDY) or RESTART for remote or local operation.

Shift Buffer Register

The shift buffer register consists of one four-bit universal bi-directional shift register, C2, that receives its input from the remote/local data selector C1. The control signals of the shift buffer register are bits C17 and C16 of the control instruction, which control the parallel loading and shift operations. The main function of the shift buffer register includes:

- Temporarily store the incoming four-bit data before shifting it into the breakpoint controller face shift register during input operation.
- During the J control command that modifies the FCR in digit mode, the shift buffer register contains the value that the FCR digit is to assume.
- The shift buffer register can also be shifted left to the breakpoint panel for display.

Output pin 15 of the shift buffer register is connected back to its left shift serial data input pin 7 to perform end-around shift left operation which saves the original data.

Control Character Register

When the character entered determines that the controlware is a control character instead of a hexadecimal digit or a terminator, the three least significant bits from the output of the remote/local data selector are loaded into the quadruple D-type flip-flop C3 by the controlware-generated

TABLE 4-4. ASCII-TO-HEXADECIMAL CONVERSION

Address$_{16}$ RR6 and RR3-RR0	ASCII Code (RR6-RR0)	Characters	PROM D1 Output (Bits 8-1)
00	0 1 1 0 0 0 0	0	C0
01	0 1 1 0 0 0 1	1	C1
02	0 1 1 0 0 1 0	2	C2
03	0 1 1 0 0 1 1	3	C3
04	0 1 1 0 1 0 0	4	C4
05	0 1 1 0 1 0 1	5	C5
06	0 1 1 0 1 1 0	6	C6
07	0 1 1 0 1 1 1	7	C7
08	0 1 1 1 0 0 0	8	C8
09	0 1 1 1 0 0 1	9	C9
0A	0 1 1 1 0 1 0	: Terminator	A0
0B	Illegal code		00
0C	Illegal code		00
0D	Illegal code		00
0E	Illegal code		00
0F	0 1 1 1 1 1 1	? Master clear	90
10	1 0 0 0 0 0 0	@ Terminator	A0
11	1 0 0 0 0 0 1	A	CA
12	1 0 0 0 0 1 0	B	CB
13	1 0 0 0 0 1 1	C	CC
14	1 0 0 0 1 0 0	D	CD
15	1 0 0 0 1 0 1	E	CE
16	1 0 0 0 1 1 0	F	CF
17	1 0 0 0 1 1 1	G Terminator	A0
18	1 0 0 1 0 0 0	H	80
19	1 0 0 1 0 0 1	I	81
1A	1 0 0 1 0 1 0	J	82
1B	1 0 0 1 0 1 1	K Control Characters	83
1C	1 0 0 1 1 0 0	L	84
1D	1 0 0 1 1 0 1	M[†]	85
1E	1 0 0 1 1 1 0	N[†]	86
1F	Illegal code		00

[†] M and N control characters are undefined.

clock, STRBCCR. The output of the control character register is decoded by decoder D5 as follows:

CC2	CC1	CC0	Control Characters
0	0	0	H
0	0	1	I
0	1	0	J
0	1	1	K
1	0	0	L
1	0	1	M (not used)
1	1	0	Open
1	1	1	Open

The output of the control character register is either gated out to the breakpoint panel for driving CONTROL CODE LED displays or transmitted to remote devices via the UART.

BREAKPOINT CONTROLLER CONTROL HARDWARE

The breakpoint controller operations are controlled by a control instruction program (controlware) that is stored in PROMs. The breakpoint controller control hardware is required mainly to execute the control instruction program and to perform other control functions as follows:

- Address the PROMs to select one of 96 control instructions.
- Decode the control instruction to generate the control signals for selectors, shift registers, etc.
- Generate micro instructions to be executed by the CPU.
- Implement miscellaneous logic circuits such as breakpoint control, controller timing chain, baud rate generator, etc.

Control Instruction Format

The panel interface controlware is contained in nine 32- by eight-bit PROMs which form 96- by 24-bit control instructions. Controlware, in this manual, is defined as a set of 24-bit control instructions that control and sequence controller operations. The control instruction is shown in figure 4-5.

The P0 field specifies the jump mode:

Bit C23 = 1	Jump if condition is true
Bit C23 = 0	Jump if condition is false

The jump condition is selected by the P1 field and the jump address is specified by the P7 field.

The P1 field consists of five bits, C18 through C22, that select a jump condition to be tested according to jump mode P0.

The P2 field consists of two bits, C16 and C17. The P2 field controls two separate functions:

- It selects one of four types of breakpoint controller generated micro instructions, referred to as LOC0 through LOC3. Refer to the paragraph on Breakpoint Controller Micro Instructions below.
- It selects one of four modes of operation for the shift buffer register:

C17	C16	Mode of Operation
0	0	Inhibit clock (NOP)
0	1	Right shift
1	0	Left shift
1	1	Parallel load

The P3 field consists of bits C14 and C15, which select one of four modes of operation for the shift register.

The P4 field consists of bits C12 and C13, which control four separate functions:

- It selects parallel input data for loading into the shift register. Either FCR00 through FCR31, MIR00 through MIR31, or breakpoint/micro-memory address can be selected. Refer to the Shift Register Input Selector description above for details.
- It selects input data for the UART. Refer to the paragraph on the Transmitter Input Data Selector above.
- It selects serial left shift input data for the breakpoint panel interface shift register and breakpoint panel shift register. Refer to the paragraph on the Left Shift Input Selector above.
- It selects the micro-memory upper/lower 16 bits and also forces display 1. Display 1 is forced by controlware to generate the appropriate micro instructions:

C13	C12	Functions
0	0	Force display 1 and set MML at STRBMMX
0	1	Force display 1 and set MMU at STRBMMX
1	0	Set MML at STRBMMX
1	1	Set MMU at STRBMMX

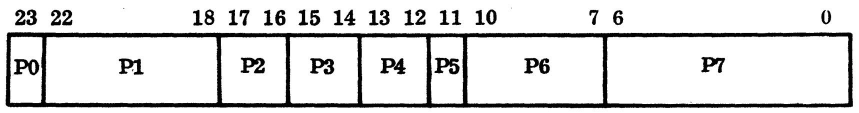


Figure 4-5. Control Instruction Format

The P5 field consists of only one bit, C11, which enables/disables the shift counter. The shift counter keeps track of how many times the shift register is shifted:

C11 = High	Enable the shift counter
C11 = Low	Disable the shift counter

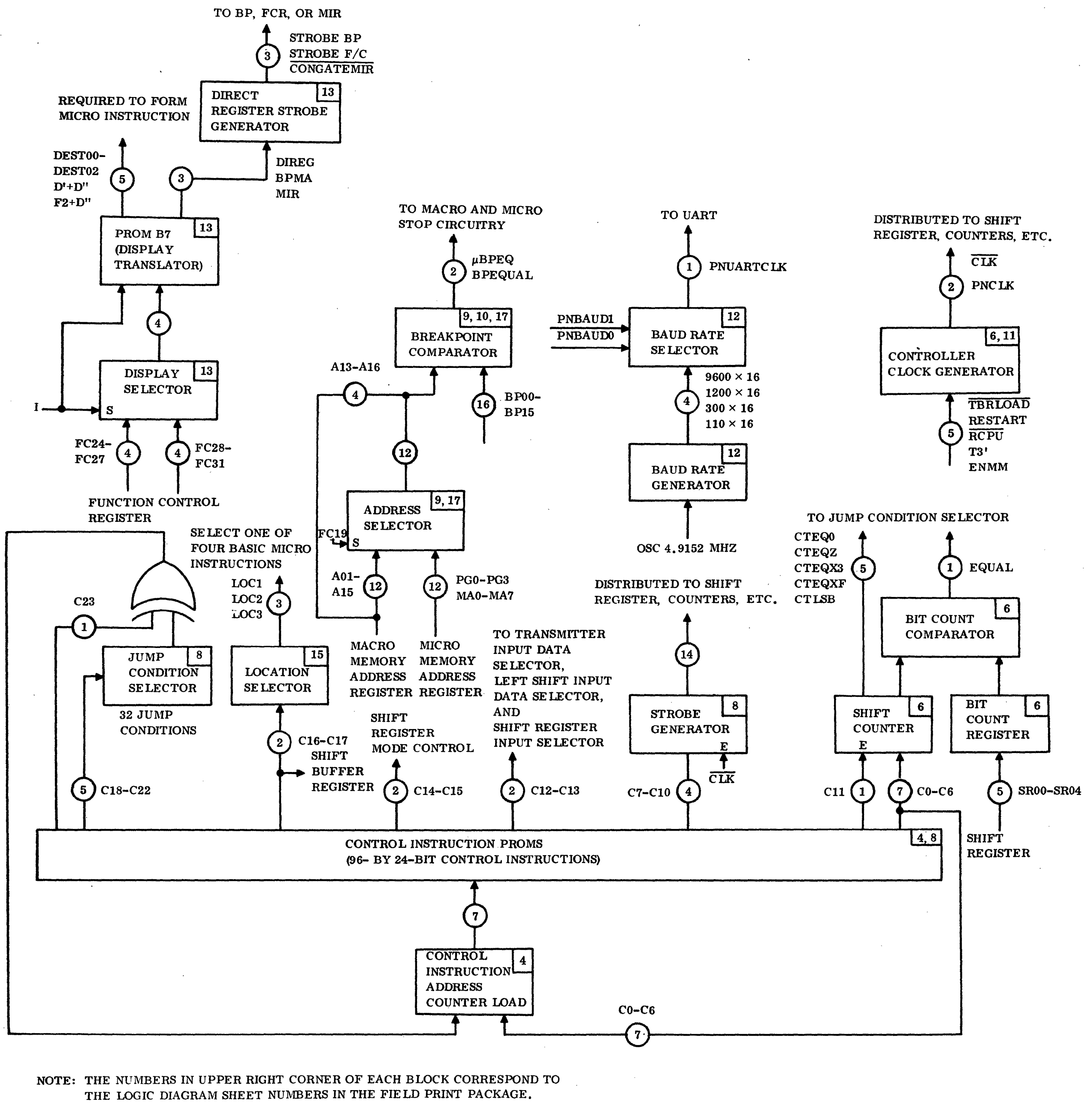
The P6 field consists of four bits, C7 through C10, that generate the strobe signals for registers, counters, etc.

The P7 field, also called the address field, consists of seven bits, C0 through C6. The P7 field can be used as follows:

- The jump address is the address of the next control instruction if the jump condition is met.
- The shift counter load value allows the shift counter to be preset to any desired value.
- The ASCII transmit character is the ASCII code of any desired character, such as the asterisk, line feed, carriage return, etc., and can be set into the P7 field and transmitted out to the remote devices via the UART.

Control Block Diagram Logic Description

Figure 4-6 shows the control block diagram of the breakpoint controller.



TO BP, FCR, OR MIR
STROBE BP
STROBE F/C
CONGATEMIR
DIRECT REGISTER STROBE GENERATOR 13
REQUIRED TO FORM MICRO INSTRUCTION
DEST00-DEST02
D'+D''
F2+D''
DIREG
BPMA
MIR
PROM B7 (DISPLAY TRANSLATOR) 13
DISPLAY SELECTOR 13
FC24-FC27
FC28-FC31
FUNCTION CONTROL REGISTER
TO MACRO AND MICRO STOP CIRCUITRY
μBPEQ
BPEQUAL
BREAKPOINT COMPARATOR 9, 10, 17
A13-A16
BP00-BP15
ADDRESS SELECTOR 9, 17
FC19
A01-A15
PG0-PG3
MA0-MA7
MACRO MEMORY ADDRESS REGISTER
MICRO MEMORY ADDRESS REGISTER
SELECT ONE OF FOUR BASIC MICRO INSTRUCTIONS
LOC1
LOC2
LOC3
C23
JUMP CONDITION SELECTOR 8
32 JUMP CONDITIONS
C18-C22
LOCATION SELECTOR 15
C16-C17
SHIFT BUFFER REGISTER
SHIFT REGISTER MODE CONTROL
C14-C15
TO TRANSMITTER INPUT DATA SELECTOR, LEFT SHIFT INPUT DATA SELECTOR, AND SHIFT REGISTER INPUT SELECTOR
C12-C13
TO UART
PNUARTCLK
PNBAUD1
PNBAUD0
BAUD RATE SELECTOR 12
9600 × 16
1200 × 16
300 × 16
110 × 16
BAUD RATE GENERATOR 12
OSC 4.9152 MHZ
DISTRIBUTED TO SHIFT REGISTER, COUNTERS, ETC.
STROBE GENERATOR 8
CLK
C7-C10
DISTRIBUTED TO SHIFT REGISTER, COUNTERS, ETC.
CLK
PNCLK
CONTROLLER CLOCK GENERATOR 6, 11
TBRLOAD
RESTART
RCPU
T3'
ENMM
TO JUMP CONDITION SELECTOR
CTEQ0
CTEQZ
CTEQX3
CTEQXF
CTLSB
EQUAL
BIT COUNT COMPARATOR 6
SHIFT COUNTER 6
BIT COUNT REGISTER 6
C11
C0-C6
SR00-SR04
SHIFT REGISTER
CONTROL INSTRUCTION PROMS (96- BY 24-BIT CONTROL INSTRUCTIONS) 4, 8
CONTROL INSTRUCTION ADDRESS COUNTER LOAD 4
C0-C6

NOTE: THE NUMBERS IN UPPER RIGHT CORNER OF EACH BLOCK CORRESPOND TO THE LOGIC DIAGRAM SHEET NUMBERS IN THE FIELD PRINT PACKAGE.

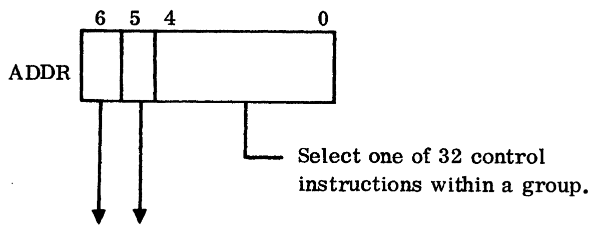
0203

Figure 4-6. Control Block Diagram

Control Instruction PROMs

The PROMs contain a total of 96 control instructions that, when executed, generate all necessary control signals for controller operation. Each PROM is organized as 32 words with eight bits per word. Therefore, a total of nine PROMs are required to form 96- by 24-bit control instructions. Figure 4-7 shows the arrangement of the PROMs. The output of PROMs J1, J2, and J3 are OR-wired to form control instruction bits C16 through C23. Similarly, the control instruction bits C8 through C15 and C0 through C7 are formed from the output of PROMs K1, K2, and K3 and L1, L2, and L3, respectively. The 96 control instructions are divided into three groups of 32 control instructions. PROMs J3, K3, and L3 form the first group of 32 control intructions, address 0_{16} through $1F_{16}$. The second group of 32 control instructions address 20_{16} through $3F_{16}$, are formed by PROMs J2, K2, and L2. The third group of 32 control instructions, address 40_{16} through $5F_{16}$, are formed by PROMs J1, K1, and L1. Table 4-5 lists the content of PROMs. The control instruction is selected by the control instruction address counter.

To select one of 96 control instructions, seven address bits, ADDR0 through ADDR6, are required. The lower five address bits, ADDR0 through ADDR4, select one of 32 control instructions within a group. The other two bits select the 32 control instruction groups by enabling the desired PROM groups:



0	0	Enable PROMs J3, K3, and L3 only.
0	1	Enable PROMs J2, K2, and L2 only.
1	X	Enable PROMs J1, K1, and L1 only.

Control Instruction Address Counter

The control instruction address counter generates address bits ADDR0 through ADDR6 to select the next control instruction. The control instruction address counter consists of two 4-bit synchronous counters, J5 and K5. The counter is cleared by the master clear signal to allow the controlware to be executed from the beginning. The address counter input is connected to the output of the PROM, bits C0 through C6. If the jump condition is met, the address counter is loaded with jump addresses C0 through C6 of the previous control instruction. If the jump condition is not met, the address counter is incremented by 1 to select the next sequential control instruction.

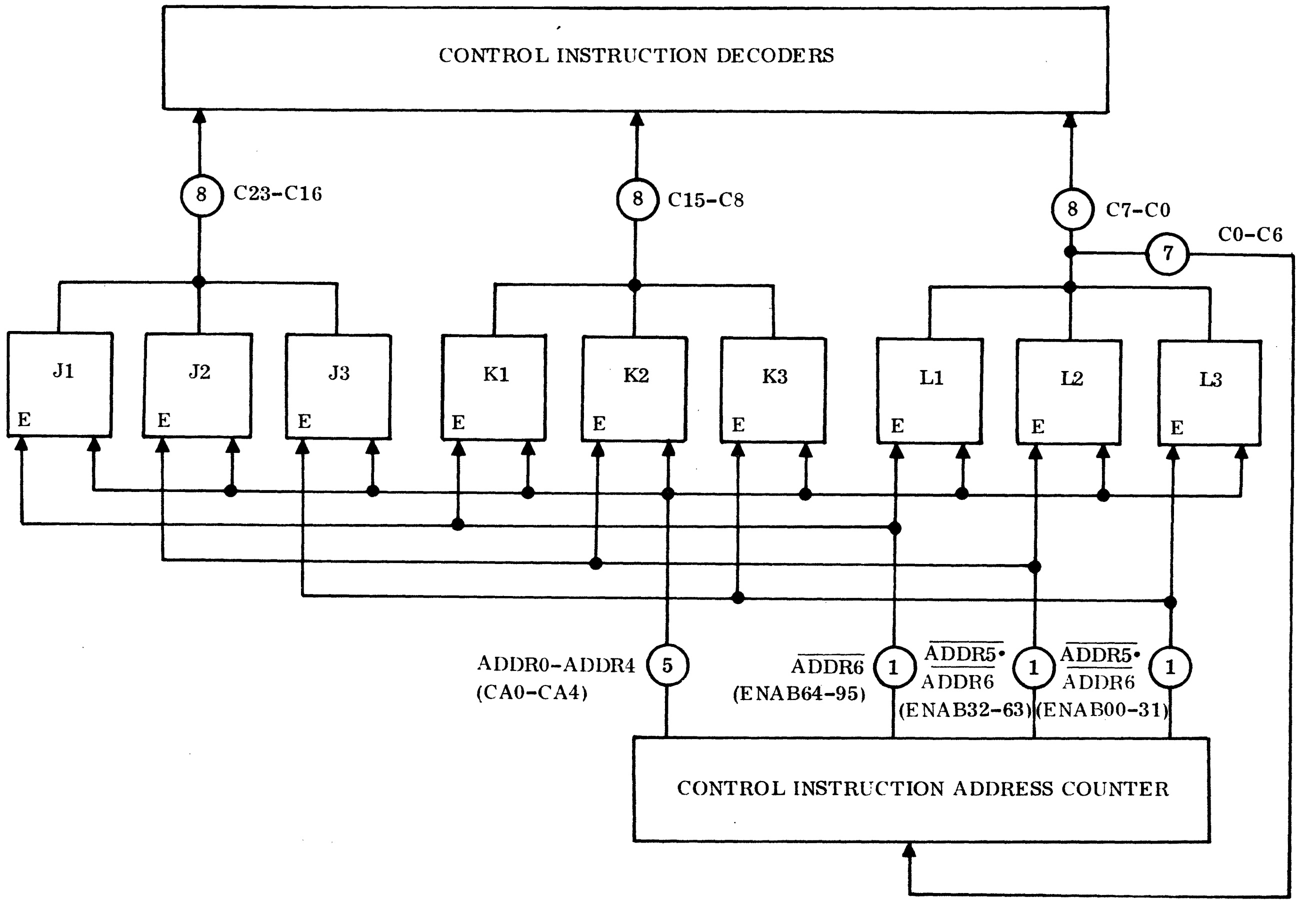

CONTROL INSTRUCTION DECODERS
8 C23-C16
8 C15-C8
8 C7-C0
7 C0-C6
J1 E
J2 E
J3 E
K1 E
K2 E
K3 E
L1 E
L2 E
L3 E
ADDR0-ADDR4 (CA0-CA4) 5
ADDR6 (ENAB64-95) 1
ADDR5• ADDR6 (ENAB32-63) 1
ADDR5• ADDR6 (ENAB00-31) 1
CONTROL INSTRUCTION ADDRESS COUNTER

Figure 4-7. PROMs Configuration to Form 96 x 24 Bit Control Instruction

TABLE 4-5. CONTROL INSTRUCTION FIELD PROGRAMMABLE READ-ONLY MEMORY

Location Address	Control Instruction Bits C23-C00								
	Word 64-95			Word 32-63			Word 0-31		
	J1	K1	L1	J2	K2	L2	J3	K3	L3
00	0C	00	42	58	1C	20	90	00	06
01	80	05	91	28	00	23	E8	00	00
02	FC	07	50	80	F3	28	30	36	12
03	80	03	44	2C	00	11	32	15	1E
04	44	A8	44	80	E3	28	7C	00	15
05	60	03	43	4C	03	28	80	00	11
06	4C	00	43	EC	00	28	84	00	08
07	82	26	43	00	07	90	80	01	08
08	44	A8	48	84	00	2B	C8	00	10
09	5C	03	4C	40	98	29	BC	00	0F
0A	80	16	43	80	06	BA	D3	00	0E
0B	81	36	4C	A0	00	2A	FC	00	0D
0C	E6	00	4E	00	21	0D	80	05	8F
0D	82	00	4E	68	00	2F	80	02	0F
0E	4C	00	53	00	21	0A	56	B8	0E
0F	80	00	52	3C	00	31	80	01	BB
10	4C	00	52	00	21	2A	3C	01	82
11	00	07	90	00	31	40	80	D3	28
12	40	A8	D2	00	21	20	68	00	11
13	5C	36	57	24	06	B6	FC	00	1D
14	80	16	55	00	11	30	58	5D	14
15	81	36	56	80	00	37	80	D3	16
16	83	03	57	00	01	40	9C	00	18
17	83	16	58	54	98	37	80	98	1B
18	80	00	59	74	00	32	B0	00	1A
19	44	A8	D9	00	21	20	82	B8	1B
1A	4C	03	5E	80	03	3B	80	88	1B
1B	60	06	5E	84	00	00	74	00	16
1C	83	16	5D	90	00	3C	80	04	11
1D	80	00	5E	80	01	3E	80	05	27
1E	44	A8	DE	E8	00	00	6C	00	40
1F	81	03	A5	80	15	00	FC	00	21

Control Instruction Decode

The control instruction is decoded to generate the control signals required for breakpoint controller operations.

P1 field decode - The P1 field of the control instruction selects a jump condition. Table 4-6 lists all the jump conditions and their descriptions. The P1 field decoder consists of two 16-to-1 multiplexers, E5 and G3. Bits C18 through C21 of the control instruction select one of 16 conditions for each multiplexer, and bit C22 enables/disables the multiplexer as follows:

C22 = Low	Enables multiplexer G3 and disables multiplexer E5
C22 = High	Enables multiplexer E5 and disables multiplexer G3

The selected jump condition is exclusively ORed with bit C23 of the control instruction to determine if a jump condition is met. A jump condition is met if:

- The condition is true and jump-if-true mode is selected (C23 = high).
- The condition is false and jump-if-false mode is selected (C23 = low).

If the jump condition is met, PNTEST4 at L7-11 goes low to load the address field P7 (bits C0 through C6) of the control instruction into the control instruction address counter. This allows the internal processor to select the control instructions required to perform a specific operation.

P2, P3, P4, and P5 field decode - These fields do not require much decoding, and therefore have been fully described in the paragraphs under Control Instruction Format above.

P6 field decode - The P6 field is decoded to generate the necessary strobe signals for registers and counters. Table 4-7 lists all the strobe signals and their functions. The P6 field consists of two 8-to-1 decoders K6 and J6. If bit C10 is low, K6 is enabled and J6 is disabled; the opposite is true if C10 is high. Note that the CLK signal is connected to allow input pin 5 to enable these decodes only when the CLK signal is low. This causes the strobe signals to have the same pulse width as the CLK signal. In addition to the strobe signals listed in table 4-7, the P6 field is also decoded by decoder H6.

C10C7	Signal Names	Function
0 0 0 1	$\overline{XENAB}$	Enable S300 to be MIR31 of shift X to left micro instruction (LOC0) if X is operator selected register.
0 1 1 0	$\overline{RDSTRB}$	Form read macro-memory code in S field of read macro-memory micro instruction.
0 1 1 1	$\overline{EXIT}$	Generate last micro instruction with M field equal to 00 to return control to the CPU.

The above signals are required in generating the micro instructions. Refer to Controller Breakpoint Micro Instructions below.

P7 field decode – The P7 field does not require any decoding. Refer to control instruction format for P7 field usage.

TABLE 4-6. JUMP CONDITIONS

C22-C18	Conditions/ Mnemonics	Descriptions
0 0 0 0 0	High	Unconditional true
0 0 0 0 1	Remote	Remote/local switch on panel is set to REMOTE.
0 0 0 1 0	N/C	Not used
0 0 0 1 1	Running	Micro running and micro mode
0 0 1 0 0	DRDY	Data is ready from the UART, or a breakpoint panel key is depressed.
0 0 1 0 1	COND.5	PINNED-out sense (not used)
0 0 1 1 0	N/C	Not used
0 0 1 1 1	EQUAL	Bit count register equals shift counter
0 1 0 0 0	FC15	Function control register bit 15 (suppress console transmit)
0 1 0 0 1	LE9	Transmitted hexadecimal digit is less than or equal to 9.
0 1 0 1 0	BPMA	BP-P/MA is selected.
0 1 0 1 1	MIR	MIR is selected.
0 1 1 0 0	$\overline{J}$	Not J control character
0 1 1 0 1	H + I	H or I control character
0 1 1 1 0	N/C	Not used
0 1 1 1 1	ERROR	Error flip-flop is set.
1 0 0 0 0	CTEQXF	Shift counter equals 31.
1 0 0 0 1	CTEQXF	Shift counter equals 15.
1 0 0 1 0	TERMCODE	Terminate code received (@, G, or :).
1 0 0 1 1	CPU 16	16-bit CPU
1 0 1 0 0	HEXRCVD	Hexadecimal digit received
1 0 1 0 1	CTEQX3	Least significant two bits of shift counter equal 11.
1 0 1 1 0	CTLSB	Least significant bit of shift counter
1 0 1 1 1	MEM	Macro memory is selected.
1 1 0 0 0	μMEM	Micro memory is selected.
1 1 0 0 1	$\overline{F2}$	File 2 is not selected.
1 1 0 1 0	NOT AUTO	Busy or not auto-display
1 1 0 1 1	DIREG	Direct registers (BP, FCR, MIR) selected
1 1 1 0 0	N/C	Not used
1 1 1 0 1	CTEQ0	Lower five bits, 0 through 4, of shift counter equal zero.
1 1 1 1 0	N/C	Not used
1 1 1 1 1	CTEQZ	Lower six bits, 0 through 5, of shift counter equal zero.

TABLE 4-7. STROBE SIGNALS

C10-C7	Strobe Signals	Location	Functions
0 0 0 0	N/C	K6-15	Not used
0 0 0 1	Test III ($\overline{\text{CLRSM214}}$)	K6-14	Clear status mode bit 214.
0 0 1 0	$\overline{\text{TBRLOAD}}$	K6-13	Load transmit buffer register of the UART.
0 0 1 1	$\overline{\text{DRRESET}}$	K6-12	Reset data ready line of the UART and set busy flip-flop high.
0 1 0 0	$\overline{\text{STRBCCR}}$	K6-11	Strobe control character register.
0 1 0 1	$\overline{\text{CLRSR}}$	K6-10	Clear shift register.
0 1 1 0	$\overline{\text{CLRCNT}}$	K6-9	Clear shift counter; strobe bit count register.
0 1 1 1	$\overline{\text{EXIT}}$	K6-7	Set M field of last micro instruction to 00 to return control to the CPU.
1 0 0 0	$\overline{\text{SSREG}}$	J6-15	Generate strobe signals for selected direct register MIR, BP, or FCR.
1 0 0 1	N/C	J6-14	Not used
1 0 1 0	$\overline{\text{STRBUL}}$	J6-13	Strobe upper/lower flip-flop K8.
1 0 1 1	$\overline{\text{SET ERROR}}$	J6-12	Set error flip-flop high.
1 1 0 0	$\overline{\text{STRBMMX}}$	J6-11	Strobe micro-memory upper/lower flip-flop.
1 1 0 1	$\overline{\text{CLRBUSY}}$	J6-10	Clear busy flip-flop and error flip-flop.
1 1 1 0	$\overline{\text{SETRCPU}}$	J6-9	Set RCPU to take over control of CPU; also disable controller free-running clock.
1 1 1 1	$\overline{\text{LOADCT}}$	J6-7	Load shift counter with address field P7, bits C0-C6.

Breakpoint Controller Micro Instructions[†]

After taking control of the processor, the breakpoint controller develops micro instructions and forces the processor to execute these instructions. These instructions are developed and loaded into the micro instruction register via the micro-memory three-state bus. There are four basic micro instructions: LOC0, LOC1, LOC2, and LOC3. They are selected by the P2 field (bits C17 and C16) of the control instruction. A combination of these micro instructions is normally required to perform breakpoint controller operations. Descriptions of the basic controller micro instructions are as follows:

LOC0 (Shift ALU X Register to Left) – This instruction shifts the ALU X register to the left one bit and is used to transfer data from the controller shift register to the X register and vice versa. The 16-bit X register and the 32-bit panel interface shift register can be considered as one 48-bit shift register.[††] The most significant bits are contained in the X register, and the least significant bits are contained in the controller shift register.

The micro instruction LOC0 contains the following:

F = 01010	(Select B input)
A = Don't care	
B = 011	(X register is the B input)
D = 110	(X register is the destination register)
S = 0000	(NOP)
C = 111010Y	(Shift destination X register to the left one bit, and enter Y in to the least significant bit position of X register.)

To transfer the contents of the X register to the lower 16 bits of the shift register and the most significant 16 bits of the shift register to the X register, the LOC0 micro instruction is executed 16 times with SR31 of the shift register selected as the Y bit of the C command. Simultaneously, the shift register is shifted left 16 times with the most significant bit of the X register (XMSB) as left shift serial input data. Since the X register is used by the breakpoint controller as a temporary storage register, the original contents of the X register must be restored at the end of the operation if X is not the selected register. The controlware generates the $\overline{\text{XENAB}}$ signal from the P6 field of the control instruction to allow either S300 (X is the selected register) or SR31 (X is not the selected register) to be MIR31 of the shift X to left micro instruction.

LOC1 (Increment micro or macro-memory pointer register by 1) – During enter/display macro memory, this micro instruction is used to increment the selected P or A register of display 1 by 1 and store it back into the selected register and macro-memory address buffer. The I register is not incremented even though this micro instruction is executed.

During enter/display micro memory, the K register is incremented by the INCK command in the C field of this micro instruction. This LOC1 micro instruction allows the macro or micro memory to be sequentially entered/displayed without re-entering the address; that is, the address is required to be input only once.

F = 11010 (ADD+) – This F code is used to increment the contents of the display 1 selected register by 1.

[†]The descriptions of these micro instructions are typical for a 16-bit machine.

[††]For a 32-bit machine, the 32-bit X register and the 32-bit panel interface shift register can be considered as one 64-bit shift register.

F = 01111 (Select A input) – This F code is used mainly to transfer the contents of the display 1 selected register to macro memory address buffer register AB.

A = The display 1 register previously selected by the operator via FCR bits FC24 through FC27

B = 001 and MIR28-29 = 11 (B source is all zeros)

D = DEST00 through DEST02 – These bits are derived from the output of PROM B7, which translates the display code into the corresponding destination register code in the D field. Refer to the description of Control Instruction Format above for PROM B7.

S = 000 (NOP)

C = Bits 28 and 29 must be 1 1 to set the B input to all zeros when required.

C = 1000101 and bit MIR19 = 0 (C' code) – Increment the K register by 1.

LOC2 (X to selected register) – This micro instruction transfers the contents of the X register to an operator selected register. During the entering operation, the input data is first shifted from the shift register to the X register. The LOC2 micro instruction is then used to store the contents of the X register in the operator selected register.

F = 01010 (Select B input)

A = 011 (X register as A input) A source is not important, since the F field always selects the B input.

B = 011 (X register is B input)

D = Operator selected destination register DEST00 through DEST02. Refer to the PROM B7 descriptions under Control Instruction Format above for details of DEST00 through DEST02.

S = 0000 (NOP) -- This indicates that the destination registers are those in the D code; e.g., ALU registers P, I, Q, F1, A, X, and F.

= 1001 (D' code) – The D' code indicates that the destination register is either a micro memory or one of status mode registers M1, M2, SM1, or SM2.

= 1011 (D" code) – The D" code indicates that the destination register is one of the double-precision registers, A*, Q*, and X*.

= 0100 (WRITE) – This code is generated when writing into macro memory is required.

= 0110 (F2WR) – This code is generated when writing data into the file 2 registers.

C = TK/O (Transform K register at selector S8 position 0) The C field must contain 0110000 and bit MIR19 = 1 (C" code). This TK/O transform allows the lowest eight bits, eight through 15, of selector S2 to be loaded into the K register.

= TN/O (Transform N register at selector S8 position 0) – The C field must contain 1000000 and bit MIR19 = 1 (C" code). This TN/O transform allows the lowest eight bits, eight through 15, of selector S2 to be loaded into the N register. Note that these instructions are the only means of loading the K and N registers.

LOC3 (Selected register to X) – During the display operation, this micro instruction is used to transfer the contents of the operator selected register to the X register. The X register is then shifted out serially to the breakpoint controller via selector S3 and displayed.

F = 01111 (A input is selected) – This F code is used when the selected register is in display 1.

= 01010 (B input is selected) – -- This F code is used when the selected register is in display 0. This F code is also required for read micro-memory operation.

A = Operator selected register in display 1 via FCR bits FC24 through FC26

B = Operator selected register in display 0 via FCR bits FC28 through FC30

D = 110 (X register is the destination register)

= 000 (NOP) – Required for read micro-memory instruction

S = 0000 (NOP) – This S code indicates that the A and B fields of the micro instruction contain the A and B codes, respectively (not A' or B' code). In other words, the operator selected register of display 1 or display 0 is a register in A or B code.

= 0111 (A' code) – The selected register of display 1 is a register in A' code; that is, SM1, SM2, M1, M2, A*, Q*, and X*.

= 1000 (B' code) – This S code is used when micro memory is the selected register and read micro-memory operation is required.

= 0011 (READ) – This S code is used to read macro memory.

C = Don't care

Table 4-8 summarizes the four basic types of micro instructions and all variations in each field of micro instruction.

The following are descriptions of important hardware required to generate the micro instructions.

MIR31 Generation – As indicated in figure 4-8, MIR31 of the shift X to left (LOC0) micro instruction is used to transfer data from the breakpoint controller shift register to the X register if MIR31 is substituted by the most significant bit, SR31, of the shift register.

If the MIR31 is substituted by the most significant bit of the X register (S300), the X register is shifted with end-around. This is required when X is the operator selected destination register.

The MIR31 is selected by 4-to-1 multiplexer H7 as follows:

C17	(X) (FETCH)	MIR31
0	0	LOC1 + SR31
0	1	S300
1	0	Low
1	1	Low

TABLE 4-8. DISPLAY CODE PROM B7

Display	Address	Selected Registers	B7 PROM Output	PROM Output (in Hexadecimal)
	0 0 0 0 0	F2	0 0 0 0 1 1 1 1	0F
	0 0 0 0 1	N	0 0 0 0 0 0 0 0	00
	0 0 0 1 0	K	0 0 0 0 0 0 0 0	00
Display 0	0 0 0 1 1	X	0 0 0 0 0 1 1 0	06
	0 0 1 0 0	Q	0 0 0 0 0 0 1 1	03
(L Function)	0 0 1 0 1	F	0 0 0 0 0 1 1 1	07
	0 0 1 1 0	F1	0 0 0 0 0 1 0 0	04
	0 0 1 1 1	MEM	0 0 0 0 0 0 0 0	00
	0 1 0 0 0	Not Used	0 0 0 0 0 0 0 0	00
	0 1 0 0 1	$\overline{\text{RTJ}}$	0 0 0 0 0 0 0 0	00
	0 1 0 1 0	Not Used	0 0 0 0 0 0 0 0	00
	0 1 0 1 1	Not Used	0 0 0 0 0 0 0 0	00
	0 1 1 0 0	MM	0 0 0 1 0 0 1 0	12
	0 1 1 0 1	Not Used	0 0 0 1 0 0 1 1	13
	0 1 1 1 0	Not Used	0 0 0 0 0 0 0 0	00
	0 1 1 1 1	Not used	0 0 0 0 0 0 0 0	00
	1 0 0 0 0	FCR	1 0 0 0 0 0 0 0	80
	1 0 0 0 1	P	0 0 0 0 0 0 0 1	01
	1 0 0 1 0	I	0 0 0 0 0 0 1 0	02
	1 0 0 1 1	Not Used	0 0 0 0 0 0 0 0	00
	1 0 1 0 0	A	0 0 0 0 0 1 0 1	05
	1 0 1 0 1	MIR	1 0 1 0 0 0 0 0	A0
	1 0 1 1 0	BP-P/MA	1 1 0 0 0 0 0 0	C0
Display 1	1 0 1 1 1	BP-P/MA Display only	1 1 1 0 0 0 0 0	E0
	1 1 0 0 0	SM1	0 0 0 1 0 1 0 1	15
(K Function)	1 1 0 0 1	M1	0 0 0 1 0 1 0 0	14
	1 1 0 1 0	SM2	0 0 0 1 0 1 1 1	17
	1 1 0 1 1	M2	0 0 0 1 0 1 1 0	16
	1 1 1 0 0	Not Used	0 0 0 0 0 0 0 0	00
	1 1 1 0 1	A*	0 0 0 1 1 1 0 1	1D
	1 1 1 1 0	X*	0 0 0 1 1 1 1 0	1E
	1 1 1 1 1	Q*	0 0 0 1 1 1 1 1	1F

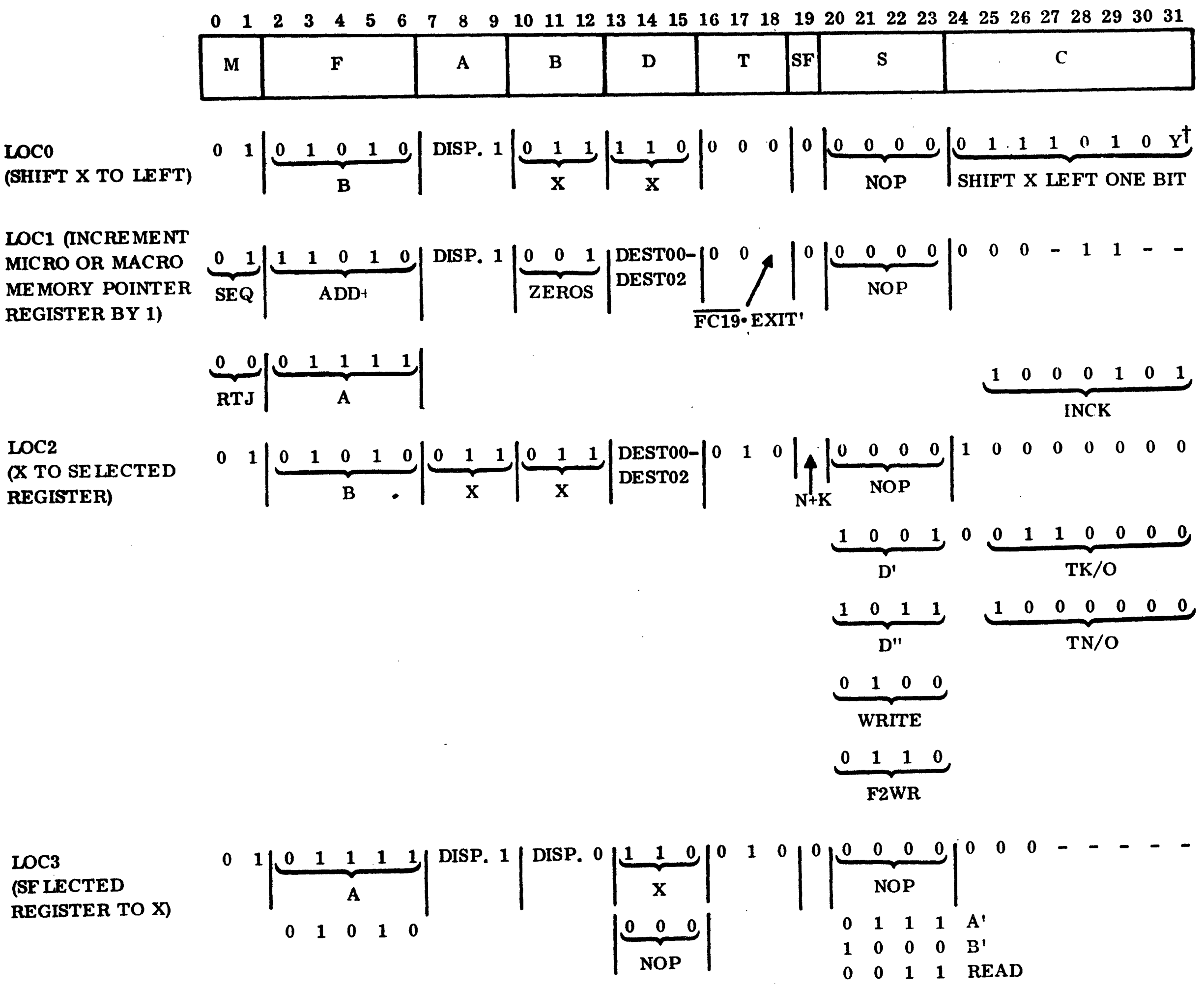
Bit C17 selects the type of micro instruction.

C17 = Low	LOC0 or LOC1
C17 = High	LOC2 or LOC3

(X)•(XENAB) selects the proper source for MIR31 depending upon if the X register is the selected destination register or not.

Display Code Translator – Table 4-2 shows the display code definition for display 1 and display 0. The selected register from either display 1 or display 0 must be translated into the corresponding code for the micro-instruction D field. The display 1 and display 0 codes consist of FCR bits FC24 through FC27 and FC28 through FC31, respectively. The display 1 and display 0 codes are first selected by 2-to-1 multiplexer B8, which is enabled only when the L or K control function is entered. Display 1 or display 0 is selected via the L signal decoded from the control character decoder.

L = Low	Selects display 1, FC24 through FC27
L = High	Selects display 0, FC28 through FC31



†WHEN ENABLED BY XENAB FROM THE CONTROLWARE, Y IS EITHER SR31 (IF X IS NOT THE SELECTED REGISTER) OR S300 (IF X IS THE SELECTED REGISTER).

0215

Figure 4-8. Breakpoint Controller Micro Instructions

Display 1 can be force-selected if bit C13 of the control instruction is low and the STRBMMX strobe signal is generated by controlware. The actual display code translation is performed by the 32- by eight-bit PROM B7. Table 4-8 lists the contents of PROM B7. The format of PROM B7 is shown in figure 4-9.

Bits 0 through 2 (DEST00 through DEST02) are used to form the destination (D) field of the micro instruction. Bits 3 and 4 (D' + D'' and F2 + D'') are used to modify the S field of the generated micro instruction. Bits 5 through 7 are used to indicate which of the direct registers, BP-P/MA, MIR, or FCR, is selected and generate the strobe signals $\overline{\text{STROBE F/C}}$, $\overline{\text{STROBE BP}}$, or $\overline{\text{CONGATEMIR}}$ accordingly.

MIR Input Data Selector – The MIR input data selector consists of 2-to-1 multiplexer DM8123 and 4-to-1 multiplexer DM8214 with three-state outputs to interface directly with the micro-memory three-state bus. The selector is enabled only when ENMM=ECPI is low (processor

7	6	5	4	3	2	1	0
DIREG	BPMA	MIR	D'+D''	F2+D''	DEST02	DEST01	DEST00

Figure 4-9. PROM B7 Format

micro memory is disabled). The selector selects either the outputs of the shift register or the breakpoint controller micro instructions to be loaded into MIR. The selector is also a part of hardware that generates the breakpoint controller micro instructions. Table 4-9 lists all the equations for each micro-instruction bit and the positions of the multiplexers that select the micro instruction type.

Shift Counter

This is a six-bit counter (J4 and K4) whose main function is to track how many times the shift register has been shifted. The shift counter consists of two four-bit synchronous counters that are enabled by bit C11 of the control instruction:

C11 = High	Enables shift counter
C11 = Low	Disables shift counter

The outputs of the counter are used to generate jump conditions. The controlware monitors the jump conditions listed below to determine if the desired number of shifts has been attained.

- CTEQX3 – The lower two bits of the counter equal 11_{16}. This jump condition is used to shift the shift register four times (one hexadecimal digit is shifted).
- CTEQXF – The lower four bits of the counter equal F_{16}. This jump condition is to be used to shift the shift register 16 or 32 times.
- CTEQ0 – The lower five bits of the shift counter equal zero.
- CTEQZ – All six bits of the shift counter equal zero.

The shift counter can be preset by the $\overline{\text{LOADCT}}$ signal to any desired value (0 through 31) via bits C0 through C5 of the control instruction. The shift counter can also be cleared by a master clear or by a $\overline{\text{CLRCNT}}$ signal from controlware.

Bit Count Register

This is a five-bit register (L10) that stores the lower five bits, SR00 through SR04, of the breakpoint controller shift register. The lower five bits of the shift register indicate the specific bits of the function control register that need to be modified. During the H or I control command, the bit count register is loaded with the bit position of the FCR bit to be cleared or set. For example in the H16: command, 16_{16} = 10110, is loaded into the bit count register.

The FCR is shifted left via the shift register, and; simultaneously, the output of the bit count register is compared with the lower five bits of the shift counter by comparator M9. When they are equal, the controlware sets or clears that FCR bit depending upon the I or H control function. During the J control command that replaces the contents of the FCR in digit mode, the digit position is first converted into the bit position by shifting to the right two places before being loaded into the bit count register. The bit count register now indicates the first bit position of the digit to be modified. For example in the J11: command, 11_{16} = 00010001 is shifted to the right two places before being loaded into the bit count register. The bit count register now contains 00100, which is the first bit position of digit 1. Only the upper three bits of the bit count register (SR02 through SR04) are compared to the upper three bits of the shift counter. The lower two bits of the bit count register are ignored and considered to be equal to the lower

TABLE 4-9. BREAKPOINT CONTROLLER MICRO INSTRUCTIONS

Micro-Memory Data Bits	RCPU = Low	RCPU = High
MM00	SR31	Low
MM01	SR30	$\overline{\text{EXIT}}$'
MM02	SR29	MEM•EXIT'
MM03	SR28	RCPU= High
MM04	SR27	$\overline{\text{MEM•EXIT}}$' LOC1+LOC3•$\overline{\text{L}}$
MM05	SR26	RCPU = High
MM06	SR25	$\overline{\text{MEM•EXIT}}$' LOC1+LOC3•$\overline{\text{L}}$
MM07	SR24	FC26•$\overline{\text{LOC2}}$
MM08	SR23	FC25+LOC2
MM09	SR22	FC24+LOC2
MM10	SR21	FC30•LOC 3
MM11	SR20	LOC•DIS01+$\overline{\text{C16}}$ †
MM16	SR15	Low
MM17	SR14	C17
MM18	SR13	$\overline{\text{FC19}}$•EXIT'
MM19	SR12	LOC2•(N+K)
MM24	SR07	C17
MM25	SR06	LOC2•NREG+LOC1•(μM+F1+ F2)+C11
MM26	SR05	C11+KREG•LOC2
MM27	SR04	C11+KREG•LOC2
MM28	SR03	LOC1•(F2+MEM)
MM29	SR02	$\overline{\text{C17}}$
MM30	SR01	Low
MM31	SR00	INST. LSB

Select Signals S1 and S0 = RCPU and FC ††

Micro Memory Data Bits	0 0	0 1	1 0 (LOC0 + LOC3)	1 1 (LOC1 + LOC2)
MM12	SR19	SR19	DIS00+$\overline{\text{LOC3}}$ †††	DIS00+$\overline{\text{LOC3}}$ †††
MM13	SR18	SR18	$\overline{\text{LOC3•μMEM}}$	DEST02
MM14	SR17	SR17	$\overline{\text{LOC3}}$•μMEM	DEST01
MM15	SR16	SR16	Low	DEST00

Select Signals S1 and S0 = C17•RCPU and RCPU'•$\overline{\text{LOC2}}$

Micro Memory Data Bits	0 0	0 1 (LOC0 or LOC1)	1 0 (LOC2)	1 1 (LOC3)
MM20	SR11	Low	D'+D"	FC31•L
MM21	SR10	Low	F2+MEM	FC27•$\overline{\text{L}}$
MM22	SR09	Low	F2+D"	FC27•$\overline{\text{L}}$+RDSTRB'• MEM
MM23	SR08	Low	D'+D"	RC27•$\overline{\text{L}}$+RDSTRB'• MEM

†DIS01 = FC29•$\overline{(\overline{\text{FC31}}\text{•}\overline{\text{FC30}}\text{•FC29•}\overline{\text{FC28}})}$ = FC29•$\overline{\text{KREG}}$

††FC = LOC1+LOC2

†††DIS00 = FC28+$\overline{\text{KREG}}$+MML

two bits of the shift counter by forcing the input pin M5-3 to be high (since $\overline{J}$ at H3-13 is low). This allows the first bit of the selected digit to be recognized, and the next four consecutive bits of FCR (one digit) to be modified once the upper three bits of the bit count register is equal to the upper three bits of the shift counter. Refer to the description of J11: operation under Typical Breakpoint Controller Operations below.

Breakpoint

There are two types of processor breakpoints: macro breakpoint and micro breakpoint. For macro breakpoint, the macro-memory address is compared with the contents of the breakpoint register; for micro breakpoint the micro-memory address is used. The macro-memory or micro-memory address is selected by 2-to-1 multiplexers C12, J13, and K11 according to FCR bit FC19 as follows:

FC19 = High (micro BP)	Select micro-memory address (PG0, PG3, MA0, MA7, and $\overline{BTLW}$)
FC19 = Low (macro BL)	Select macro-memory address (A01 through A12)

The macro and micro breakpoint conditions are selected by FC16 and FC17 via 4-to-1 multiplexer J7:

FC17	FC16	Macro Breakpoint (J7-1)	Micro Breakpoint (J17-14)
0	0	Low: breakpoint not selected	Breakpoint not selected
0	1	SELGETMAK: instruction reference breakpoint	($\overline{BTLW}$ + BP12)· µBPEQ
1	0	CPU WRITE: store operand breakpoint†	µBPEQ
1	1	High: all reference breakpoint	µBPEQ

The contents of the breakpoint register are compared to the macro- or micro-memory address by 4-bit magnitude comparators: C10, C11, H13, and K9. The comparator is enabled by the $\overline{RCPU}$ at K9 3 when it is high. This indicates that the breakpoint operation is performed only when the breakpoint controller is not in control of the processing element. The µBPEQ or BPEQUAL signal is generated when the micro-memory address is equal to the breakpoint register.

For macro breakpoint, if the breakpoint register is equal to the macro-memory address and the breakpoint select conditions are met, the processor is stopped. If bit FC18 of the FCR is set, it causes the $\overline{SETSM104}$ signals at M6-2 to generate a macro breakpoint interrupt (SM104) instead of a stop. For micro breakpoint, if the micro-memory address PG0 through PG3, MA0 through MA7, and the upper/lower micro-instruction selection signal $\overline{BTLW}$ are equal to breakpoint register bits BP00 through BP12 and the combination of FCR bits 17 and 16 is not zero, then a micro stop occurs ($\overline{MICROSTOP}$ at M8-8 goes to low). If FCR bit FC17 is set, then comparison of breakpoint bit 12 and the upper/lower micro-instruction selection signal $\overline{BTLW}$ is not required.

Breakpoint Controller Timing Chain

The breakpoint controller clock signal ($\overline{CLK}$) that is used for controller shift registers, breakpoint panel shift registers, control instruction address counter, and shift counter is comprised of two independent clocks: the breakpoint controller interface internal clock and the processor T3 clock. Figure 4-10 shows the breakpoint controller clock.

Breakpoint Controller Internal Clock – The controller internal clock is used for execution of internal operations independently of the processor.

The controller internal clock generator consists of the dual retriggerable one-shot, A4, that generates approximately 120 nanoseconds symmetrical clock pulses. The clock generator can be stopped and started by controlling the triggering input signal to the first one-shot as follows:

- The breakpoint controller internal clock generator is free-running only when the $\overline{RCPU}$ at G4-4 is high, which indicates internal operation. However, when breakpoint controller operation is executed in conjunction with the processor ($\overline{RCPU}$ is low), the trigger input fed back from the second one-shot is disabled, causing the clock generator to stop. The clock generator is restarted after the breakpoint controller exits from the processor.
- The breakpoint controller clock generator can also be stopped and restarted via the $\overline{TBRLOAD}$ signal. During local operation (the remote/local switch of the breakpoint panel is in the LOCAL position), the universal asychronous receiver/transmitter (UART) is disabled. This allows the strobe transmitter buffer register signal $\overline{TBRLOAD}$ of the UART to be used for debouncing the breakpoint panel keys. If any key of the breakpoint panel is depressed, the $\overline{TBRLOAD}$ signal is generated to set flip-flop B6-9 to high. This in turn stops the internal clock to allow the key to be debounced. The $\overline{TBRLOAD}$ signal also triggers one-shot C5 to produce approximately a 15 millisecond $\overline{RESTART}$ signal. The trailing edge of the $\overline{RESTART}$ signal resets flip-flop B6-9 and allows the internal clock generator to run again. The controller internal clock generator is also stopped by master clear.

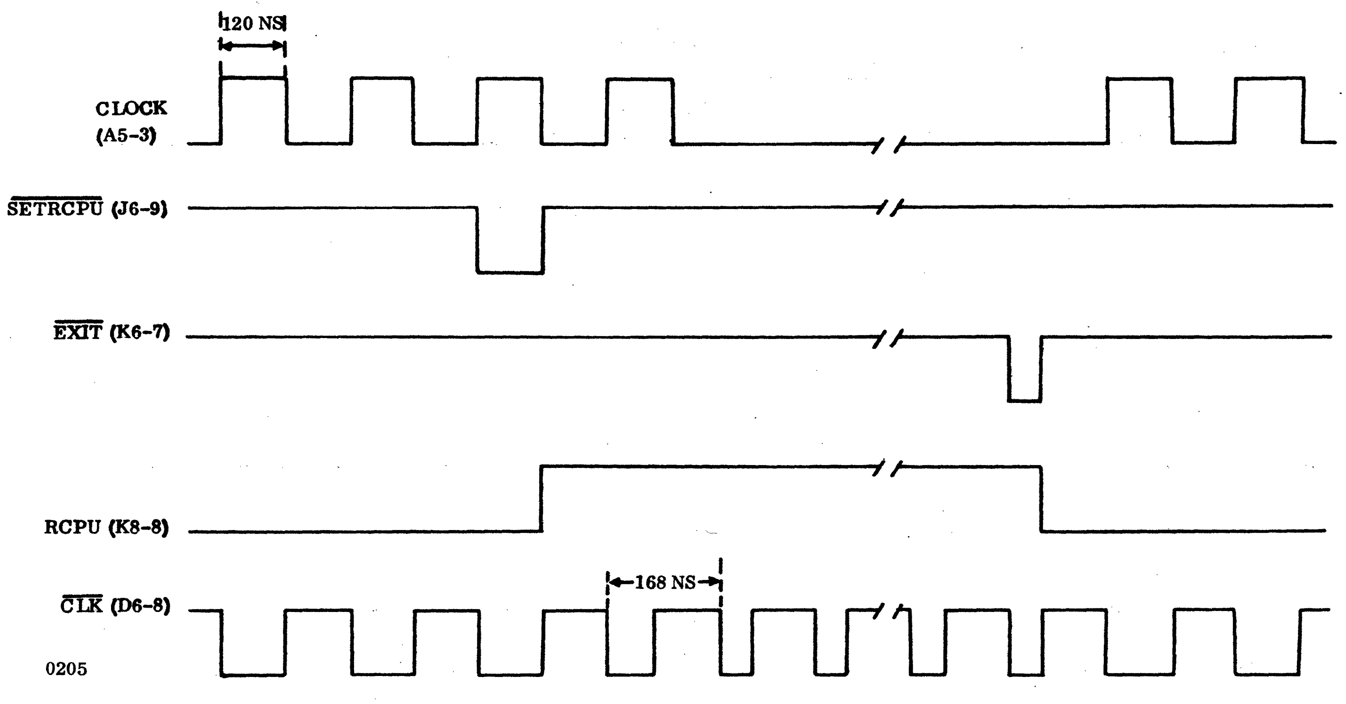
Processor T3 Clock – The T3 clock is derived from the processor odd/even clock generator. It is required to synchronize the operation in the breakpoint controller with the processor operation; for example, shift the shift register once every time the processor executes the micro instruction LOC0 that shifts the X register to the left one time. The T3 clock is enabled only when the breakpoint controller operation is executed in conjunction with the processor. During this time, the controller internal clock is stopped. The equation that generates the controller timing chain, $\overline{CLK}$, is as follows:

$$\overline{CLK} = \overline{(ENMM+RCPU)\cdot CLOCK+T3'\cdot(ENMM+\overline{RCPU})}$$

Breakpoint Controller and Processor Interface Signals

Before the breakpoint controller can force the processor to execute the controller micro instructions required for a specific operation, it must take over processor control by generating $\overline{SETSM214}$ or $\overline{INT14}$ (see figures 4-2 and 4-3).

†In basic MOS processor, the CPUWRITE signal is replaced by CPUBKPWRT signal from ADDR/CONTROL board.



120 NS
CLOCK (A5-3)
$\overline{\text{SETRCPU}}$ (J6-9)
$\overline{\text{EXIT}}$ (K6-7)
RCPU (K8-8)
168 NS
$\overline{\text{CLK}}$ (D6-8)
0205

Figure 4-10. Panel Interface Clock

$\overline{\text{SETSM214}}$ – If the processor is in micro mode, the breakpoint controller must generate the SETSM214 signal, which prepares the processor before it gives control to the breakpoint controller:

$\overline{\text{SETSM214}} = \overline{\text{FC19·SET RCPU}}$

FC19 = 1 indicates that the processor is in micro mode. The SETRCPU signal is generated by controlware whenever it determines that the processor is required to change/fetch the selected register.

$\overline{\text{INT14}}$ – If the processor is in macro mode, $\overline{\text{INT14}}$ is generated to signal the emulator that breakpoint controller operation is requested. Once $\overline{\text{INT14}}$ is recognized (the breakpoint panel requests CPU interrupt), the emulator processes this interrupt to prepare the processor before giving up control to the breakpoint controller. $\overline{\text{INT14}}$ at K8-8 is set low by SETRCPU from controlware. After the desired operation is completed, the breakpoint controller exits the CPU by generating the $\overline{\text{EXIT}}$ signal which in fact clears the $\overline{\text{INT14}}$ signal. The RCPU signal is also generated as a complement of $\overline{\text{INT14}}$ by the same logic circuit.

Go and Stop

The breakpoint controller has the capability to start and stop the processor. Depending upon the processor mode of operation, micro or macro, the corresponding micro or macro go or stop signal is generated via the control function, H or I. The processor mode of operation is determined by FCR bit FC19.

FC19 = High	Micro breakpoint, step, go and stop
FC19 = Low	Macro breakpoint, step, go and stop

$\overline{\text{CONSTART}}$ – The $\overline{\text{CONSTART}}$ signal is generated at flip-flop J8-5 to cause a micro go if either:

- It is entered regardless of FCR bit FC19.
- Auto-display is enabled (i.e., FC10 is set) and any terminator, :, @ , or G, is entered with no characters preceeding it.
- The $\overline{\text{CONSTART}}$ signal is generated by the breakpoint controller $\overline{\text{SETRCPU}}$ signal. This allows the breakpoint controller to start the processor when it is in micro mode and to initially stop.

The above three conditions form the clock input for flip-flop J8-3.

$$\text{J8-3} = \overline{\text{SETRCPU+(I+AUTO)·CTEQZ·}}\ \overline{\text{TERMCODE·DRREST·}\overline{\text{J}}}$$

The $\overline{\text{CONSTART}}$ signal is sent to the control 1 module to allow the processor main clock generator to run.

$\overline{\text{EXGO}}$ – The $\overline{\text{EXGO}}$ signal is generated at F6-11 to cause a macro go. $\overline{\text{EXGO}}$ is sent to the status mode interrupt module to set the SM215 bit. When the emulator detects SM215 being set, it causes a macro go. $\overline{\text{EXGO}}$ is generated under the same condition 1 or 2 of $\overline{\text{CONSTART}}$ generation but FCR bit FC19 must be cleared.

$$\overline{\text{EXGO}} = \overline{\text{[(I+AUTO)·CTEQZ·TERMCODE·DRRESET·J]·}}\ \overline{\text{FC19}}$$

$\overline{\text{CONSTOP}}$ – The $\overline{\text{CONSTOP}}$ signal is generated at M8-8 to cause a micro stop if FCR FC19 is set and either:

- The H: command is entered.
- The breakpoint controller exits from the processor.

These two conditions form the clock input for flip-flop J8-11.

$$J8\text{-}11 = [(H\cdot NOT\ AUTO)\cdot CTEQZ\cdot TERMCODE\cdot DRRESET\cdot\overline{J}]+EXIT$$

If either of the above conditions is true, flip-flop J8-3 goes low to generate $\overline{CONSTOP}$.

- The micro step operation executes:

  $$J9\text{-}10 = \overline{FC20\cdot\overline{RCPU}}$$

- The micro-memory address is equal to the contents of the breakpoint register, and the micro breakpoint condition is met.

All of the above conditions are gated by FC19. The $\overline{CONSTOP}$ signal is sent to the control 1 module to stop the odd/even time generator. Refer to the control 1 module description in the Basic Micro-Programmable Processor Hardware Maintenance Manual for more details.

$\overline{EXSTOP}$ – The $\overline{EXSTOP}$ signal is generated at F6-6 to cause a macro stop. $\overline{EXSTOP}$ is sent to the status mode interrupt module to clear the SM215 bit. When the emulator detects that the SM215 is clear, it causes a macro stop. Macro stop is generated when FCR bit FC19 is clear and either:

- The H: command is entered:

  $$J9\text{-}4 = \overline{(H\cdot NOT\ AUTO)\cdot CTEQZ\cdot TERMCODE\cdot \overline{DRRESET}\cdot\overline{J}}$$

- The macro step operation executes:

  $$J9\text{-}3 = \overline{FC20\cdot CPUEDS\cdot SEL\ GETMAK}$$

- The macro-memory address is equal to the contents of the breakpoint register, and the macro breakpoint condition is met.

All of the above conditions are gated by $\overline{FC19}$.

Baud Rate Generator

The baud rate is selectable through a baud rate selection switch in the I/O-TTY controller.

The baud rate generator provides four different UART baud rates: 9600, 1200, 300, and 110 baud. The receiver/transmitter clock frequency is 16 times the desired baud rate; therefore, the actual clock frequencies corresponding to the above baud rates are 9,600 times 16, 1,200 times 16, 300 times 16, and 110 times 16, respectively. The baud rate generator consists mainly of three four-bit synchronous binary counters, M1, M2, and M3, which are cascaded by using the positive overflow carry output of the first counter to enable the successive cascaded stage. This arrangement forms a 12-bit frequency divider that divides the 4.9152 MHz oscillator signal from the control 2 module down to the desired clock frequencies.

UART Clock Frequencies	4.9152 MHz Input Clock Divided by	Location
9,600 x 16	32	M2 - 14
1,200 x 16	256	M2 - 11
300 x 16	1024	M3 - 13

A clock frequency of 75 times 16 Hz would be generated at M3-11 if straight division were used. However, by allowing the counter to skip 653 ($28D_{16}$) clock pulses twice, a 110 times 16 clock frequency is generated at M3-11.

These four baud rates are selectable by PN BAUD1 and PN BAUD0 from the I/O-TTY controller in the 4-to-1 multiplexer M4.

PN BAUD1	PN BAUD0	UART Clock Frequencies
0	0	110 x 16 = 1,760 Hz
0	1	300 x 16 = 4,800 Hz
1	0	1,200 x 16 = 19,200 Hz
1	1	9,600 x 16 = 153,600 Hz

When the 110 baud rate is selected, the SBSELECT signal at M4-15 is high to select two stop bits to be transmitted by the UART. For the other baud rates, 9,600, 1,200, and 300, a single stop bit is selected.

BREAKPOINT CONTROLLER CONTROLWARE

The controlware is defined in this manual as a set of 96- by 24-bit control instructions that sequences and controls controller operations when executed. Refer to Control Instruction Format. The panel controlware is preprogrammed in the control instruction PROMs. An actual listing is included in appendix A for reference.

Controlware Listing Description

The controlware listing includes the definitions of all control instruction fields, P1 through P7. The listing format is explained using an example from the listing.

```
                        P1     P2  P3  P4  P5  P6      P7
                        ↓      ↓   ↓   ↓   ↓   ↓       ↓
CHKREM   J TRUE   REMOTE    ,  ,   ,   ,   ,   ,   CHARDY   JUMP IF   REMOTE
0684
0008
```

Where:

- CHKREM is the name of the subroutine.
- J TRUE is the jump mode as indicated by bit 23 of the control instruction being set.
- REMOTE is the jump condition selected by the P1 field. Note that the blank space between commas represents a field of the control instruction as shown above.
- CHARDY is the name of the jump location when the jump condition is met.
- JUMP IF REMOTE is the general remark of the control instruction.
- The most significant two hexadecimal digits, 06_{16}, represent the control instruction address in the PROMs.
- The six hexadecimal digits, 840008_{16}, represent the control instruction itself.

Figure 4-11 shows the controlware bit configuration.

There are four main types of control instructions:

- Jump if condition true (J TRUE) – Execute the control instruction specified by address field P7 (C0 through C6) if the jump condition is true. If not, select the next sequential control instruction.
- Jump if condition false (J FALSE) – Execute the control instruction specified by address field P7 if the jump condition is false. If not, select the next sequential control instruction.
- Unconditional jump (JUMP) – Unconditionally execute the control instruction specified by address field P7. The jump condition is always satisfied in this case.
- Sequential execution (SEQ) – The next sequential control instruction is always executed. In this case, address field P7 may contain the address of the next sequential control instruction or transmit data to the UART.

Basic Controlware Operations

The breakpoint controller controlware performs three basic operations: input, change/fetch, and transmit.

- Input – The controlware is continuously executed in the idle loop consisting of the first two control instructions. As soon as data is ready from the UART or a breakpoint panel switch is depressed, the controlware goes into input operation to process the incoming data. For a typical control command, the controlware first stores the control character code at the control character register if it is a legal code, then stores the hexadecimal data (four-bit data) one after another at the shift register while monitoring for the terminator code. When the terminator code is received, the controlware goes into the change/fetch operation.
- Change/fetch – In this operation, the breakpoint controller requests the CPU (if required) and generates the micro instructions required for the operation indicated by the controller control command. These micro instructions are executed by the CPU. If the desired operation is an enter operation, the contents of the selected register are first loaded with new data (change operation), and then they are fetched out (fetch operation) to the X register and the shift register (SR). However, if the operation is a display operation, the change operation is omitted and only the fetch operation is performed. After the fetch operation is completed, the breakpoint controller exits and returns control to the CPU before going into the transmit operation.
- Transmit – The transmit operation is used to send the response to the remote device or breakpoint panel for display.

Figures 4-12, 4-13, and 4-14 show flow charts of the input operation, the change/fetch operation, and the transmit operation, respectively.

Typical Breakpoint Controller Operations

The following typical breakpoint controller operations are chosen to illustrate the combined operation of the hardware and controlware. The operations are explained in terms of flow charts and general remarks for each step. Refer to the controlware listing, the panel interface data path block diagram (figure 4-4), and control block diagram (figure 4-6) as necessary.

J11: Control Command

The J11: control command selects the P register by changing digit 1 of the FCR to a value of 1_{16}.

For clarity, assume that the FCR originally contains 12345678_{16}. Digit 1, containing a value of 2_{16}, is changed to a value of 1_{16}. Also assume that a remote device (programmers console) is used and that FCR bit FC15 is not set (the console transmit is not suppressed).

The J11: operation is explained using three major flow charts. The input operation is shown in figure 4-15, the change/fetch operation in figure 4-16, and the transmit operation in figure 4-17. The input operation is divided into three separate flow charts: enter J control character, enter hexadecimal digit 1, and enter terminator :.

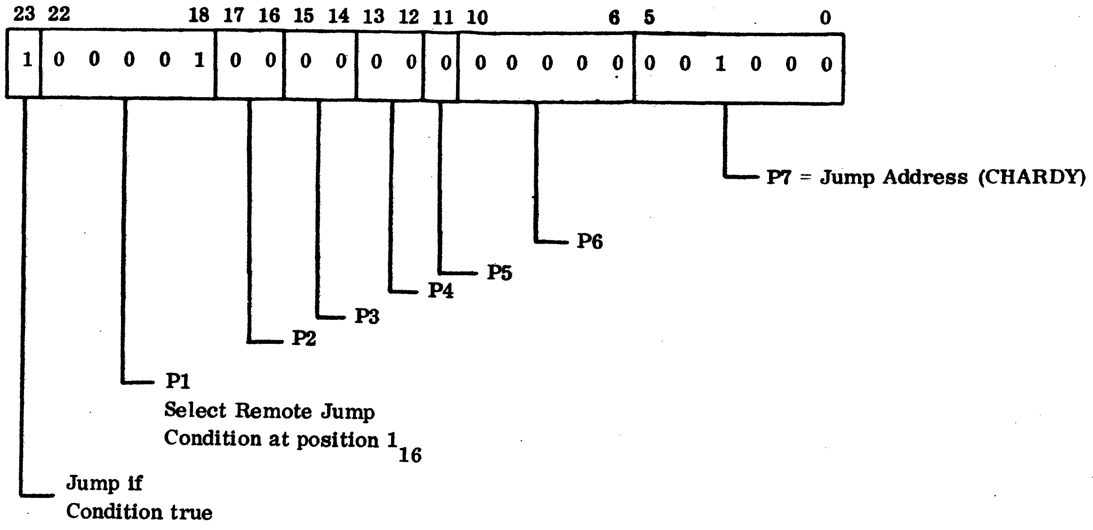


Figure 4-11. Controlware Bit Configuration

K1234: Command

Since the P register is already selected (FC27 through FC24 equal to 0001) in the above example, the next logical operation would be to load the P register with 1234_{16}. The K1234: command is required for this operation. The input and transmit operations are quite similar to the above example. Figure 4-18 is the change/fetch operation for the enter P with 1234 command (K1234:).

BREAKPOINT PANEL

The breakpoint panel is the device used for the man/machine interface via the breakpoint controller` module. The breakpoint panel provides a means for entering breakpoint controller control commands and displaying the response. In panel mode, the breakpoint panel is capable of doing any operation that any remote device can do. The breakpoint panel switches and display LEDs are shown and described in figure 2-1 and table 2-1.

Figure 4-19 shows the functional block diagram of the breakpoint panel The breakpoint panel consists of:

- Momentary switches for control character and data selection
- Switch encoder
- Panel shift register
- Control code display
- Data display and UPPER Indicator display
- LOCAL/REMOTE switch
- MASTER CLEAR switch

Only the switch encoder and panel shift register of the block diagram are described in detail here. The rest of the breakpoint panel block diagram is either self-explanatory or has been described in table 2-1.

SWITCH ENCODER

The switch encoder is used to encode any depressed control character switch or data switch into the corresponding binary code to be used by the breakpoint controller. The switch encoder consists of three priority encoder chips: U10, U12, and U13. The control character switches are encoded by U10. The data switches are encoded by U12 and U13. The truth table of these eight-input priority encoders is included for reference (table 4-10). The control character encoder U10 is always enabled except when a control character switch is being pressed. This makes the control character switches higher priority than the data switches when they are pressed simultaneously.

TABLE 4-10. EIGHT-INPUT PRIORITY ENCODER TRUTH TABLE

Inputs									Outputs				
5	4	3	2	1	13	12	11	10	6	7	9	14	15
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	L	X	X	X	X	X	X	X	L	L	L	L	H
L	H	L	X	X	X	X	X	X	L	L	H	L	H
L	H	H	L	X	X	X	X	X	L	H	L	L	H
L	H	H	H	L	X	X	X	X	L	H	H	L	H
L	H	H	H	H	L	X	X	X	H	L	L	L	H
L	H	H	H	H	H	L	X	X	H	L	H	L	H
L	H	H	H	H	H	H	L	X	H	H	L	L	H
L	H	H	H	H	H	H	H	L	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L

The PNLCC signal at U10-14 goes low if any of the control character switches are pressed to indicate to the breakpoint controller that a control character is entered and not hexadecimal data. If any of the breakpoint panel switches are pressed, the PNLKYDEP signal at U13-15 goes active. This signal indicates to the breakpoint controller that data is ready. (Breakpoint panel data ready is processed in the same manner as the data ready line from the UART during remote operation.) The outputs of priority encoders U10, U12, and U3 are ORed together to form the binary codes PNHEX00 through PNHEX03. Table 4-11 shows the binary code of the breakpoint panel switches. The terminator code received condition is generated in the breakpoint controller by ANDing PNLCC and PNHEX00 through PNHEX03.

BREAKPOINT PANEL SHIFT REGISTER

The breakpoint panel shift register converts serial data PNLSLI from the breakpoint controller to parallel format. The 16-bit output of the shift register is used to drive the data display LEDs. The breakpoint panel shift register consists of four universal bidirectional shift register chips: U5, U6, U7, and U8. The shift register mode of operation is controlled by single bit C15 (PNLSHIFT) of the control instruction; therefore, it only operates in left shift and hold modes. The shift register is cleared by the PNLCLR signal from controlware. The clock PNLCLK is the same breakpoint controller CLK as the breakpoint controller shift register for synchronization (see Breakpoint Controller Internal Clock above). The serial left-shift input data, PNLSLI, is the most significant bit (SR31) of the breakpoint controller shift register. This allows the contents of the breakpoint controller shift register to be displayed. LEDs DS02 and DS17 correspond to the most and least significant data bits of the breakpoint controller shift register. PNLSLI is also used to drive the UPPER indicator when the output of breakpoint controller upper flip-flop K8 is selected. LED DS01 is the upper indicator.

CONTROL CODE DISPLAY

Control code LEDs DS18 through DS20 are driven by signals PNLITE0 through PNLITE2. These signals are the outputs (CC0 through CC2) of the control character register gated by the output of the error flip-flop H1 in the breakpoint controller. Therefore, when there is an error, all control code LEDs are illuminated.

GLOSSARY

Table 4-12 is a tabulation of abbreviations and mnemonics used throughout the text and figures of this manual.

TABLE 4-11. BINARY CODE OF BREAKPOINT PANEL SWITCHES

Switches	PNLCC	PNHEX03	PNHEX02	PNHEX01	PNHEX00
0	1	0	0	0	0
1	1	0	0	0	1
2	1	0	0	1	0
3	1	0	0	1	1
4	1	0	1	0	0
5	1	0	1	0	1
6	1	0	1	1	0
7	1	0	1	1	1
8	1	1	0	0	0
9	1	1	0	0	1
A	1	1	0	1	0
B	1	1	0	1	1
C	1	1	1	0	0
D	1	1	1	0	1
E	1	1	1	1	0
F	1	1	1	1	1
H	0	1	0	0	0
I	0	1	0	0	1
J	0	1	0	1	0
K	0	1	0	1	1
L	0	1	1	0	0
M (not used)	0	1	1	0	1
N (not used)	0	1	1	1	0
:	0	1	1	1	1

TABLE 4-12. GLOSSARY OF TERMS

Term	Description	Term	Description
A (A*)	A register (A register double-precision)	DRDY	Data ready
ADDR	Address bits (00 through 15)	DREADY	Data ready
ALU	Arithmetic logical unit	DRRESET	Data ready reset
A01-A15	Address bits (0 through 15) macro memory	ECPI	Enable breakpoint controller micro memory bus
BKPT	Breakpoint	ENABFC	Enable function control
BP	Breakpoint	ENDARND	Shift the shift register end-around
BPEQUAL	Breakpoint equal	ENMM	Enable micro memory
BPMA	Breakpoint memory address registers	ENTERHEX	Enter hexadecimal number
BP-P/MA	Breakpoint page/memory address registers	EXGO	External go
BP00-BP15	Breakpoint bits (00 through 15)	EXSTOP	External stop
BTLW	Bit test lower/upper micro instruction	F	F register
CCR	Control character register	F1	File no. 1
CHAR	Character	FCR	Function control register
CHARDY	Character ready	FC15	Function code bit (15)
CHKREM	Check remote	F/F	Flip-flop
CKBPMA	Check breakpoint memory address	GATEAB	Gate CPU memory address register clock
CKFC16	Check function control bit 16	GATET	Gate test bit
CKF2	Check field F2	HEXRCVD	Hexadecimal digit received
CKZERO	Check for zero	I	I register
CLK	Clock	INCK	Increment K register
CLR	Clear	INT	Interrupt
CLRCT	Clear counter	K	K register
CLRUL	Clear upper limit	LED	Light emitting diode
CONGATEMIR	Control gate micro-instruction register	LE9	Less than or equal to 9
CONSTART	Control start	LOC0,1,2,3	Location number (0 through 3)
CONSTOP	Control stop	LSB	Least significant bit
CPU	Central processing unit	MEM	Memory
CPUEDS	Central processing unit early data strobe	MIR	Micro-instruction register
CT	Counter	MM	Micro memory
CTEQXF	Count equals 15 hexadecimal	MML	Micro-memory lower 16 bits (16 through 31)
CTEQX3	Count equals 3 hexadecimal	MMU	Micro-memory upper 16 bits (00 through 15)
CTEQZ	Count equals zero	MM00-MM31	Micro-memory instruction bits (00 through 31)
CTEQ0	Count equals zero	MSB	Most significant bit
CTLSB	Counter least significant bit	N	N register
C00-C23	Control instruction bits (firmware)	P	P register
DECMAC	Decrement micro memory address counter	P/MA	Page/memory address registers
DES00-DES02	Destination selection code bits (00-02)		
DIREG	Direct registers selected		
DIS	Display		
DIS0MU	Display 0 memory upper		
DIS1ML	Display 1 memory lower		

TABLE 4-12. GLOSSARY OF TERMS (Contd)

Term	Description	Term	Description
PNBAUD0	Panel baud rate selection bit 0	SSREG	Strobe select for direct register
PNBAUD1	Panel baud rate selection bit 1	STRBCCR	Strobe control character register
PNHEX00-PNHEX15	Panel hexadecimal number bits (00 through 15)	STRBMMX	Strobe micro memory bit
PNKEYDEP	Panel key depressed	STRBUL	Strobe upper/lower
PNLCLK	Panel clock	TB	Test bit
PNLCLR	Panel clear	TBRLOAD	Load transmit buffer
PNLOCAL	Panel local	TERM	Terminate
PNSLI	Panel left shift serial input	TERMCODE	Terminate code
PNTEST	Panel test	TMRCVD	Terminate received
PNUARTCLK	Panel UART clock	TRO	Transmit register output
PROM	Programmed read-only memory	TTL	Transistor transistor logic
Q	Q register	UART	Universal asynchronous receiver transmitter
Q*	Q register double precision	U/L F/F	Upper/lower flip-flop
RCPU	Request central processing unit	X	X register
RDR	Data ready	X*	X register double-precision
RDSTRB	Read stroke	XCR	Transmit carriage return
RNI	Read next interrupt	XENAB	X register enable
RRI	Receive register input	XHEXGR9	Transmit
RR0-RR7	Receive register data bits (0 through 7)	XHEXLE9	Transmit hexadecimal number less than or equal to 9
RS232	Type of transmission line characteristics	XLF	Transmit line feed
RTJ	Return jump	XMIT	Transmit
SBSELECT	Strobe select	XMITCCR	Transmit control character
SELGETMAK	Select GET MAK	XMITFCR	Transmit function control register
SETCPU	Set central processing unit	XMSB	X register most significant bit
SR	Shift register	XSPACE	Transmit space
SRL	Shift the shift register left	μBPEQ	Micro memory address equals breakpoint register
SRLOAD	Load shift register		
SRR	Shift the shift register right		
SR00-SR31	Shift register bits (00 through 31)		

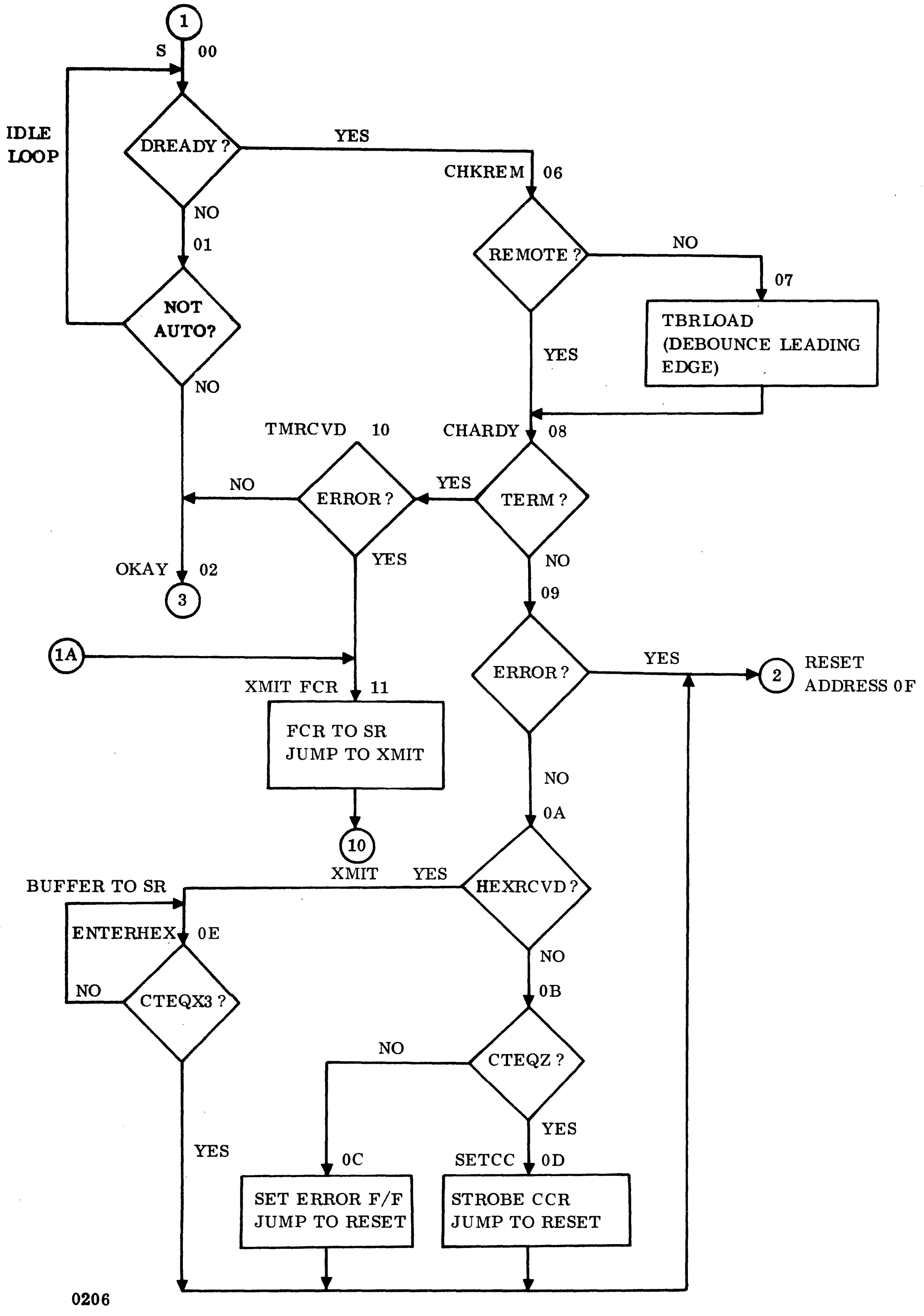


Figure 4-12. Input Operation Flow Chart (Sheet 1 of 2)

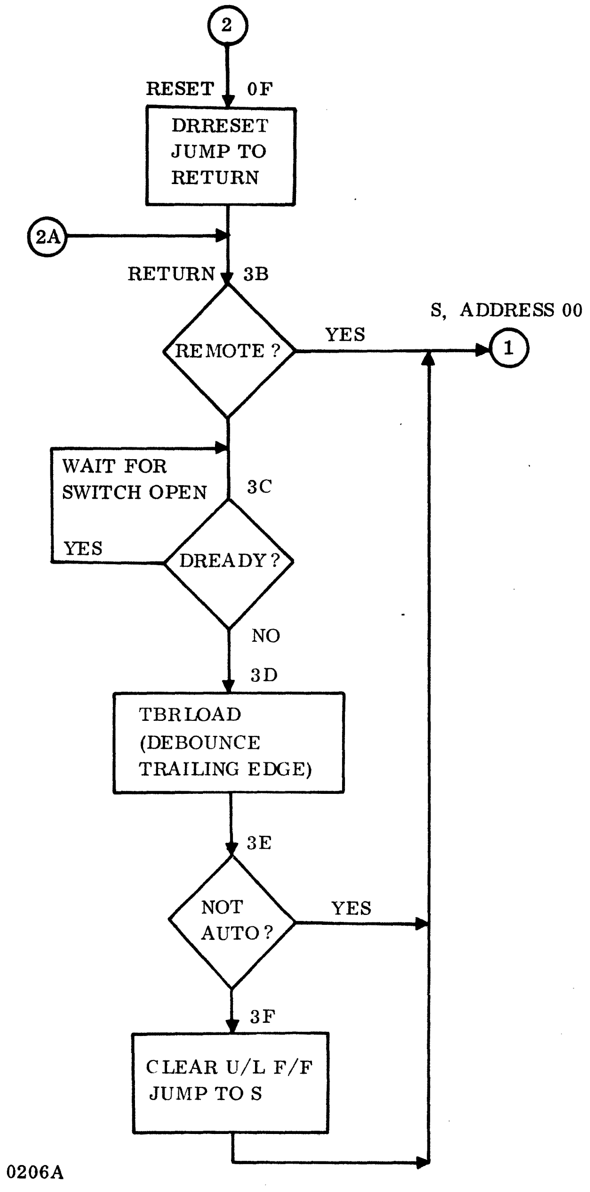


Figure 4-12. Input Operation Flow Chart (Sheet 2 of 2)

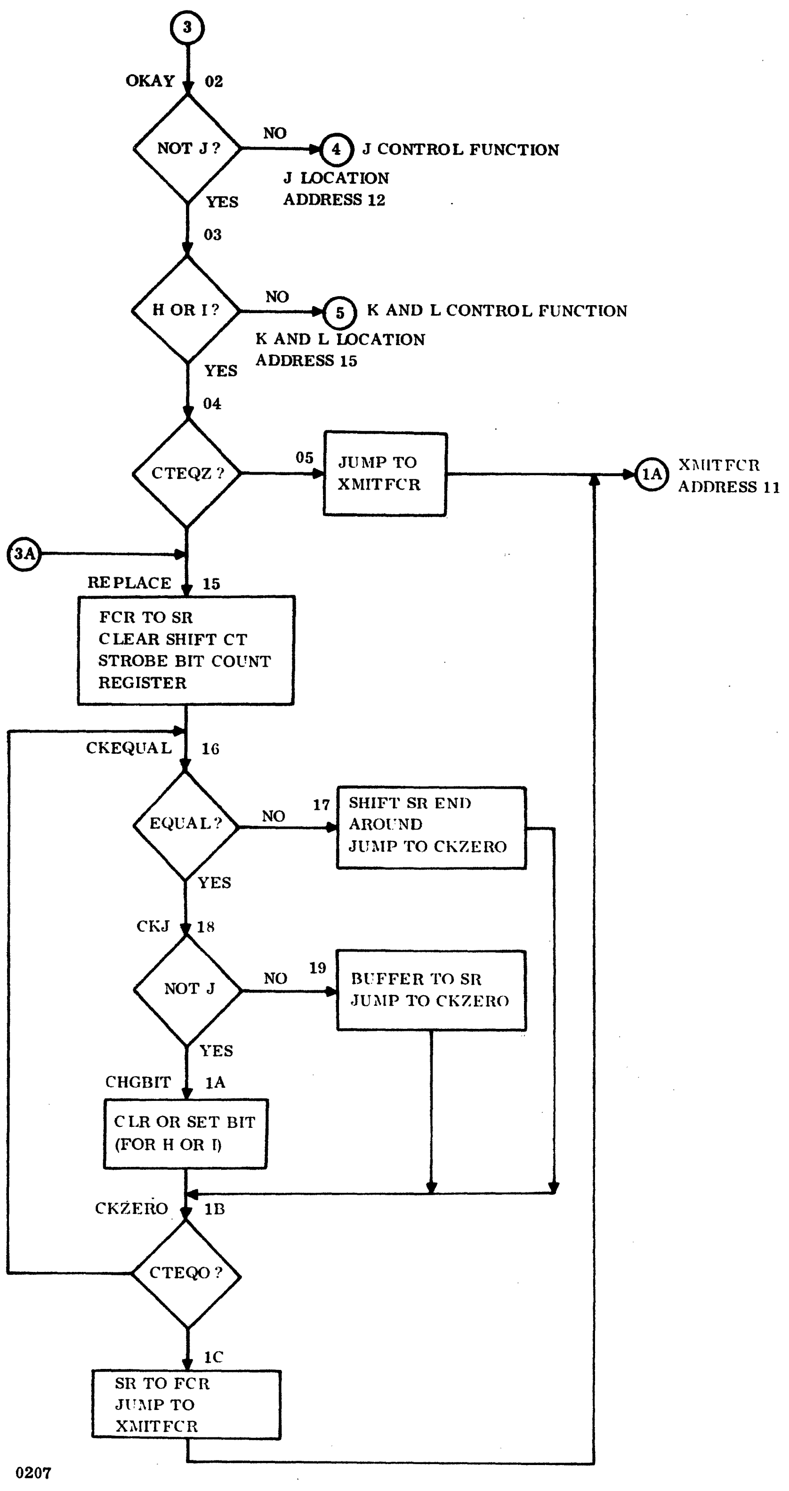
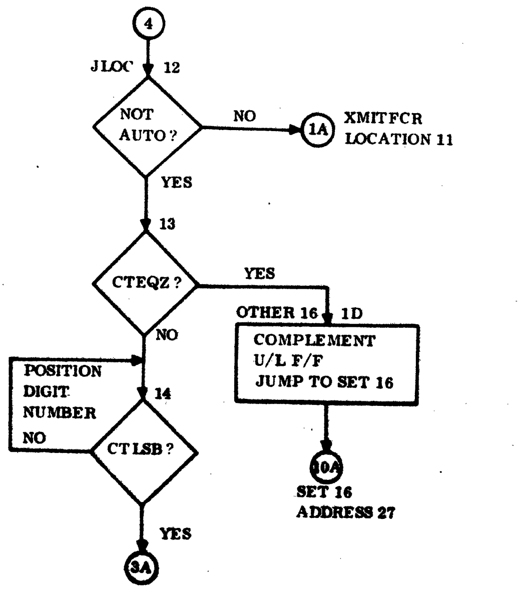
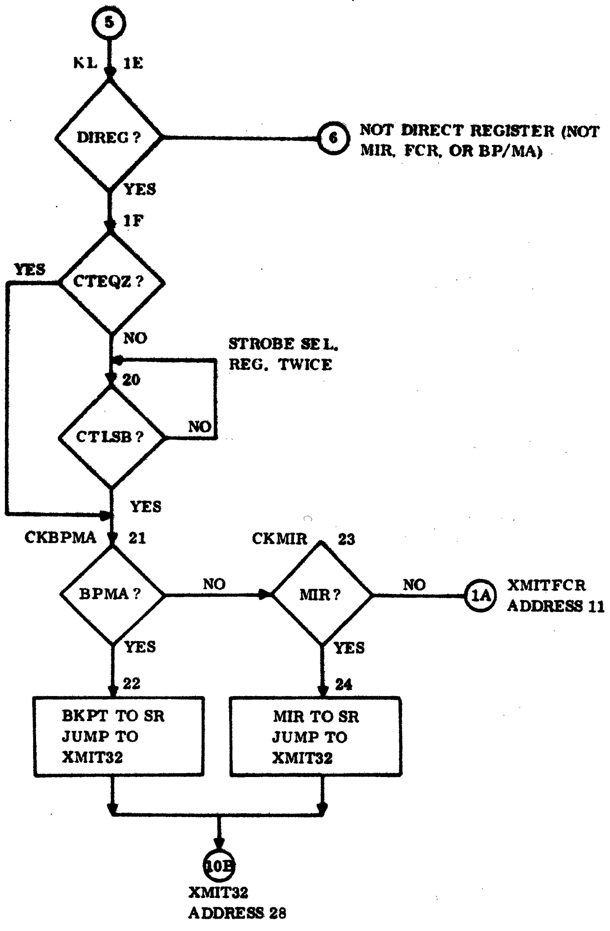


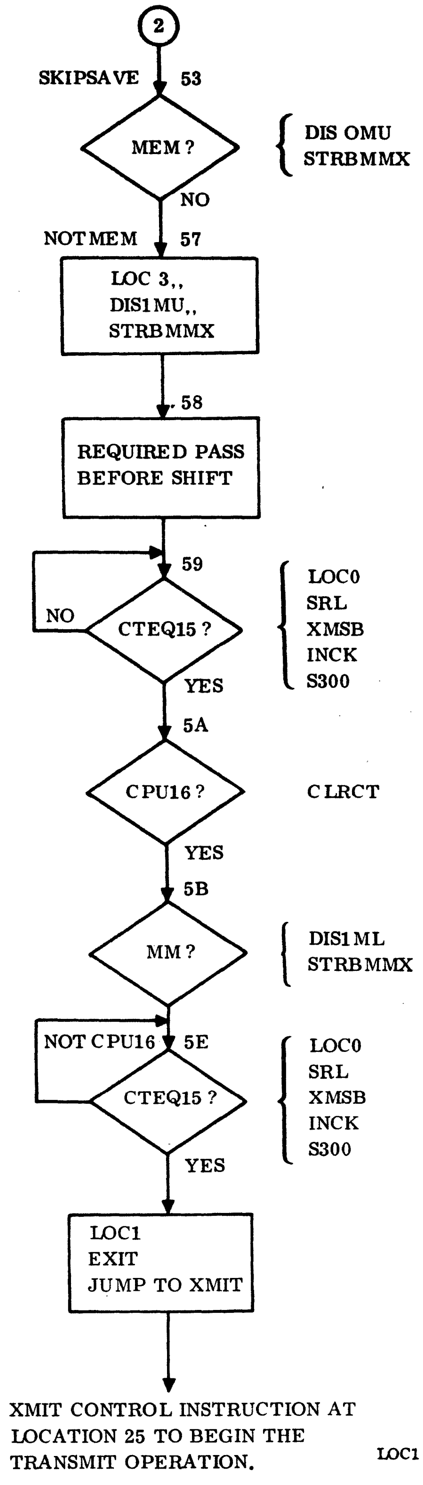
Figure 4-13. Change/Fetch Operation Flow Chart (Sheet 1 of 5)





0207A

Figure 4-13. Change/Fetch Operation Flow Chart (Sheet 2 of 5)

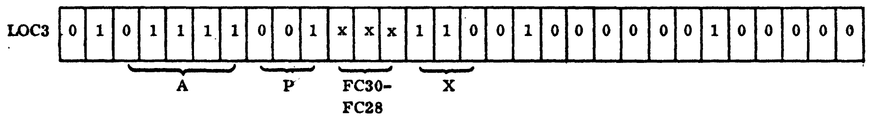


0212B

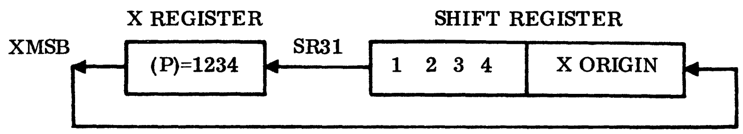
REMARKS

Set MMU at STRBMMX.

The LOC3 micro instruction transfers the contents of the P register to the X register.

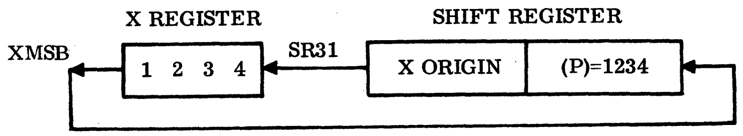


The X register and shift register now contain:



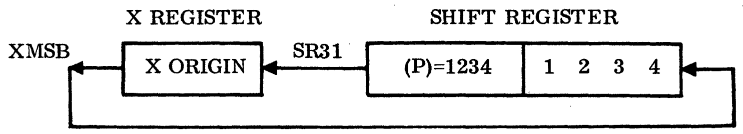
Also force display 1.

Shift 16 bits.



Clear the shift counter.

Shift 16 bits.



The S300 from this control instruction also clears the SM214.

The breakpoint controller exits and returns control to the CPU by generating the EXIT strobe, which sets the M field of this LOC1 micro instruction to 00 (RTJ mode). M field equal to 00 causes the ENMM signal in the control 1 module to be high, disabling the controller and enabling the micro memory to the MIR. The T field of the micro instruction is equal to 000 for micro mode.

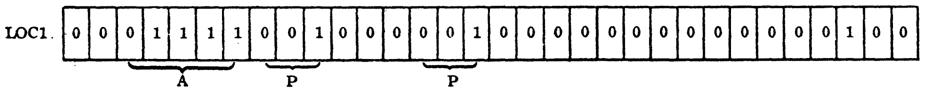
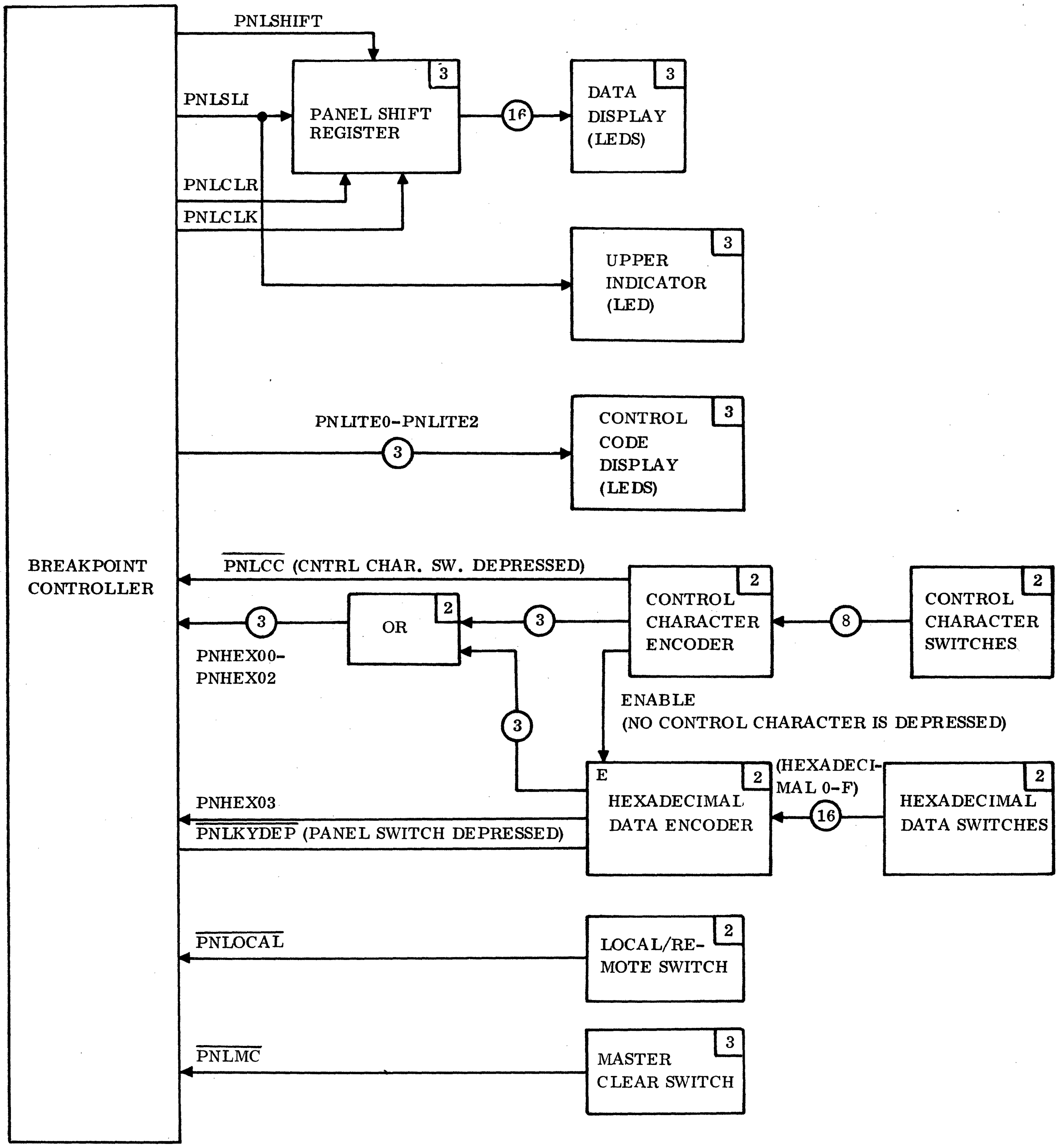


Figure 4-18. Enter P with 1234 command (K1234:), Change/Fetch Operation (Sheet 3 of 3)



NOTE: THE NUMBER IN THE UPPER RIGHT CORNER OF EACH BLOCK CORRESPONDS TO THE LOGIC DIAGRAM SHEET NUMBER IN THE FIELD PRINT PACKAGE.

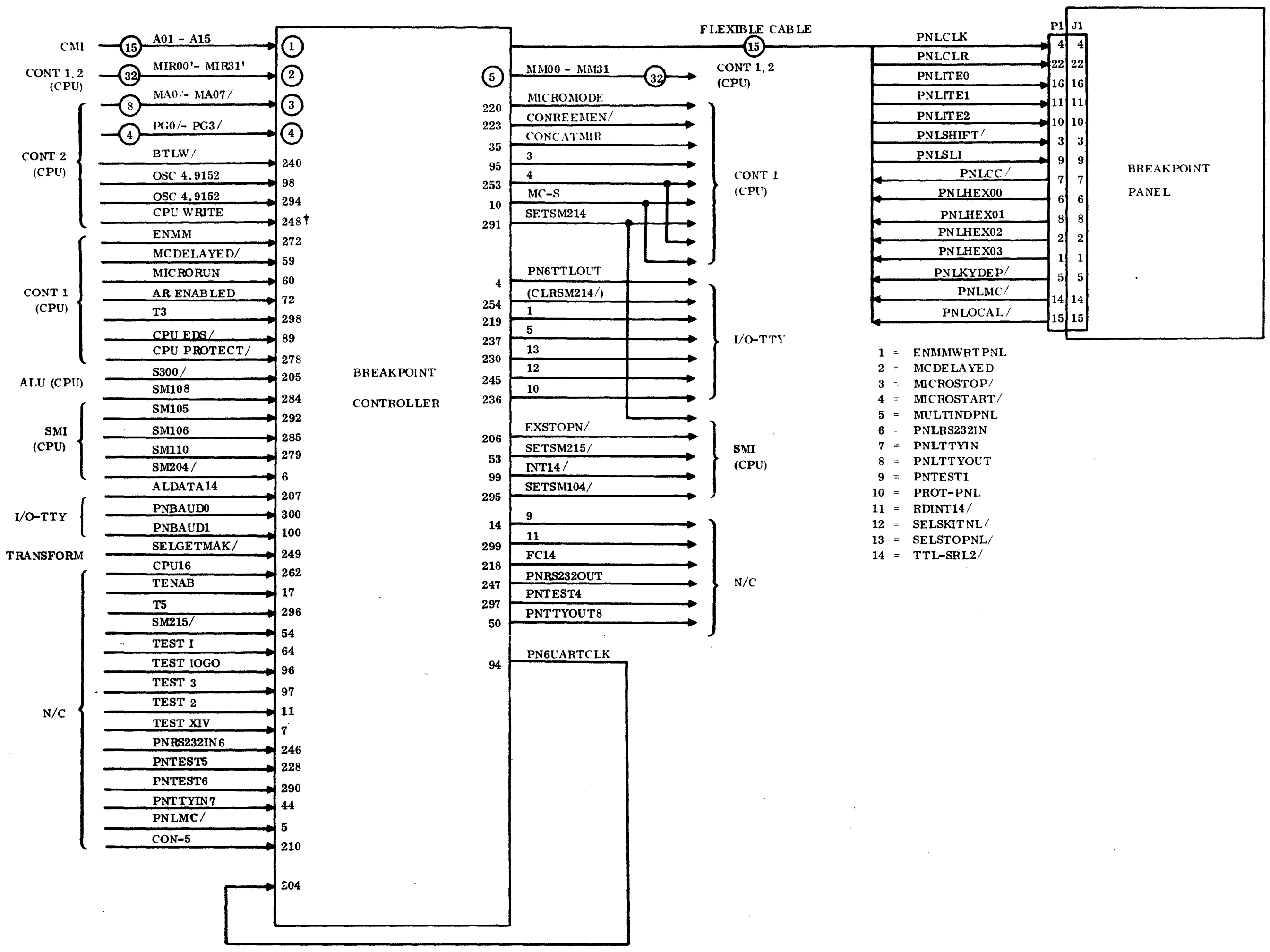
Figure 4-19. Breakpoint Panel Functional Block Diagram

LOGIC DIAGRAMS

The logic diagrams for the breakpoint controller and the breakpoint panel are located in the field print package.

INTERCONNECTING SIGNAL DIAGRAM

Figure 5-1 shows all the interface signals between the processor, breakpoint controller, and the breakpoint panel. Signal names and logic board pin numbers are included. This diagram should be useful for troubleshooting to the board level.



† IN BASIC MOS PROCESSOR, THE CPUWRITE SIGNAL IS REPLACED BY THE CPUBKPWRT SIGNAL FROM THE ADDR/ CONTROL BOARD.

0214

Figure 5-1. Interconnecting Diagram (Sheet 1 of 2)

PIN NUMBERS

① A01 - A15

A01	271
A02	273
A03	275
A04	277
A05	263
A06	265
A07	267
A08	269
A09	225
A10	229
A11	260
A12	258
A13	259
A14	233
A15	234

③ $\overline{\text{MA0}}$ - $\overline{\text{MA7}}$

$\overline{\text{MA0}}$	256
$\overline{\text{MA1}}$	19
$\overline{\text{MA2}}$	87
$\overline{\text{MA3}}$	92
$\overline{\text{MA4}}$	281
$\overline{\text{MA5}}$	283
$\overline{\text{MA6}}$	280
$\overline{\text{MA7}}$	86

④ $\overline{\text{PG0}}$ - $\overline{\text{PG3}}$

$\overline{\text{PG0}}$	36
$\overline{\text{PG1}}$	39
$\overline{\text{PG2}}$	38
$\overline{\text{PG3}}$	29

② MIR00' - MIR31'

MIR00'	213	MIR21'	241
MIR01'	209	MIR22'	43
MIR02'	212	MIR23'	244
MIR03'	12	MIR24'	62
MIR04'	9	MIR25'	61
MIR05'	214	MIR26'	287
MIR06'	16	MIR27'	286
MIR07'	215	MIR28'	288
MIR08'	250	MIR29'	293
MIR09'	252	MIR30'	75
MIR10'	231	MIR31'	88
MIR11'	232		
MIR12'	261		
MIR13'	268		
MIR14'	270		
MIR15'	69		
MIR16'	21		
MIR17'	15		
MIR18'	224		
MIR19'	8		
MIR20'	243		

⑤ MM00 - MM31

MM00	28	MM21	46
MM01	31	MM22	45
MM02	27	MM23	42
MM03	26	MM24	77
MM04	25	MM25	78
MM05	24	MM26	79
MM06	23	MM27	80
MM07	22	MM28	81
MM08	55	MM29	82
MM09	56	MM30	83
MM10	57	MM31	84
MM11	58		
MM12	63		
MM13	65		
MM14	66		
MM15	70		
MM16	40		
MM17	41		
MM18	49		
MM19	48		
MM20	47		

Figure 5-1. Interconnecting Diagram (Sheet 2 of 2)

MAINTENANCE 6

On-site maintenance (emergency or preventive maintenance) is limited to the diagnosis of a malfunction to the subassembly level. Repair is effected by replacing the faulty interface card or panel subassembly with a previously tested spare.

SPARES TESTING

All spare subassemblies must be tested upon receipt and retested annually.

PREVENTIVE MAINTENANCE

No preventive maintenance is required for this equipment.

CONTROLWARE LISTING

```
JOB,PANEL2,LOGUE
1700 MASS STORAGE OPERATING SYSTEM VERSION 4.2   DATE OF RUN: 10/16/75  SYSTEM ID:  1700 # 3 -- 32K SYSTEM    (09/06/75)
```

```
PPPPPPPPPPPP       AAAAAAAAAAA      NNN       NNN     EEEEEEEEEEEEE     LLL                22222222222
PPPPPPPPPPPPP     AAAAAAAAAAAAA     NNN       NNN     EEEEEEEEEEEEE     LLL               2222222222222
PPPPPPPPPPPPP     AAAAAAAAAAAAA     NNN       NNN     EEEEEEEEEEEEE     LLL               2222222222222
PPP       PPP     AAA       AAA     NNNN      NNN     EEE               LLL               222       222
PPP       PPP     AAA       AAA     NNNNN     NNN     EEE               LLL                         222
PPP       PPP     AAA       AAA     NNNNNN    NNN     EEE               LLL                       222
PPPPPPPPPPPPP     AAAAAAAAAAAAA     NNN NNN   NNN     EEEEEEEEEEEE      LLL                      222
PPPPPPPPPPPPP     AAAAAAAAAAAAA     NNN  NNN  NNN     EEEEEEEEEEEE      LLL                     222
PPPPPPPPPPPP      AAAAAAAAAAAAA     NNN   NNN NNN     EEEEEEEEEEEE      LLL                    222
PPP               AAA       AAA     NNN    NNNNNN     EEE               LLL                   222
PPP               AAA       AAA     NNN     NNNNN     EEE               LLL                 222
PPP               AAA       AAA     NNN      NNNN     EEE               LLL               222
PPP               AAA       AAA     NNN       NNN     EEEEEEEEEEEEE     LLLLLLLLLLLLL     2222222222222
PPP               AAA       AAA     NNN       NNN     EEEEEEEEEEEEE     LLLLLLLLLLLLL     2222222222222
PPP               AAA       AAA     NNN       NNN     EEEEEEEEEEEEE     LLLLLLLLLLLLL     2222222222222
```

*ASSEM

```
PANEL                PAGE   1               DATE: 10/16/75

                  NAM       PANEL
         SEQ      MAC       P2,P3,P4,P5,P6,P7    SEQUENTIAL EXECUTION
                  IFC       ,EQ,'P7'
          VFD X8/*/2-S,N1/1,N5/0,X2/'P2',
          VFD X2/'P3',,X2/'P4',,X1/'P5',,X4/'P6',,X7/*/2+1-S
                  EIF
                  IFC       ,NE,'P7'
          VFD X8/*/2-S,N6/0,X2/'P2',
          VFD X2/'P3',,X2/'P4',,X1/'P5',,X4/'P6',,X7/'P7'
                  EIF
                  EMC
         JUMP     MAC       P2,P3,P4,P5,P6,P7    UNCONDITIONAL JUMP
          VFD X8/*/2-S,N1/1,N5/0,X2/'P2',
          VFD X2/'P3',,X2/'P4',,X1/'P5',,X4/'P6',,X7/'P7'/2-S
                  EMC
         JTRUE    MAC       P1,P2,P3,P4,P5,P6,P7 JUMP IF CONDITION TRUE
          VFD X8/*/2-S,N1/1,X5/'P1',,X2/'P2',
          VFD X2/'P3',,X2/'P4',,X1/'P5',,X4/'P6',,X7/'P7'/2-S
                  EMC
         JFALSE   MAC       P1,P2,P3,P4,P5,P6,P7 JUMP IF CONDITION FALSE
          VFD X8/*/2-S,N1/0,X5/'P1',,X2/'P2',
          VFD X2/'P3',,X2/'P4',,X1/'P5',,X4/'P6',,X7/'P7'/2-S
                  EMC
```

```
                *        TEST EQUATFS (P1 ACTUALS)
          0000           EQU       TRUE(00)            UNCONDITIONAL ONE
          0001           EQU       REMOTE(01)          LOCAL/REMOTE SW SET TO REMOTE
          0002           EQU       SWITCHUP(02)        NO PANEL SWITCH IS DEPRESSED
          0003           EQU       RUNNING(03)         MICRO-RUNNING AND MICRO-MODE
          0004           EQU       DREADY(04)          DATA IS READY FROM UART
          0005           EQU       CONDS(05)           PINNED-OUT SENSE (NOT USED)
          0007           EQU       EQUAL(07)           COUNTER EQUALS BIT COUNT REG
          0008           EQU       FC15(08)            BIT 15 OF FUNCTION CONTROL REG
          0009           EQU       LE9(09)             XMIT DIGIT IS LT OR EQ 9
          000A           EQU       BPMA(10)            BP-MA SELECTED
          000B           EQU       MIR(11)             MIR SELECTED
          000C           EQU       NOTJ(12)            NOT J
          000D           EQU       HORI(13)            H + I
          000F           EQU       ERROR(15)           ERROR F/F
          0010           EQU       CTEQ31(16)          COUNTER EQUALS 31
          0011           EQU       CTEQ15(17)          COUNTER EQUALS 15
          0012           EQU       TERM(18)            TERMINATE CODE RECEIVED (COLON)
          0013           EQU       CPU16(19)           16 BIT CPU
          0014           EQU       HEXRCVD(20)         HEX DIGIT RECEIVED
          0015           EQU       CTEQX3(21)          LOWER 2 BITS OF COUNTER EQUAL 11
          0016           EQU       CTLSB(22)           LSB OF COUNTER
          0017           EQU       MEM(23)             MAIN MEMORY SELECTED
          0018           EQU       MM(24)              MICRO MEMORY SELECTED
          0019           EQU       NOTF2(25)           FILE 2 SELECTED NOT
          001A           EQU       NOTAUTO(26)         BUSY OR NOT AUTO-DISPLAY
          001B           EQU       DIREG(27)           D  CT REGISTER (ON THIS MOD)
          001D           EQU       CTEQ0(29)           C   TER EQUALS ZERO (BITS 0-4)
          001F           EQU       CTEQZ(31)           C   TER EQUALS ZERO (BITS 0-5)

                *        LOC COUNT AND SHIFT BUFFER ENABLE   UATES (P2 ACTUALS)
          0000           EQU       LOC0(0)             SH  T X REG LEFT BY 1
          0001           EQU       LOC1(1)             DI   + 1 TO DIS1, AR
          0002           EQU       LOC2(2)             X    SELECTED REG
          0003           EQU       LOC3(3)             SE   ED REG TO X

          0000           EQU       SBNOP(0)            SHIF.  UFFER UNAFFECTED
          0001           EQU       SBR(1)              SHIFT BUFFER SHIFTED RIGHT
          0002           EQU       SBL(2)              SHIFT BUFFER SHIFTED LEFT
          0003           EQU       SBLOAD(3)           SHIFT BUFFER LOADED

                *        SHIFT REGISTER ENABLE EQUATES (P3 ACTUALS)
          0000           EQU       SRNOP(0)            SHIFT REGISTER UNAFFECTED
          0001           EQU       SRR(1)              SHIFT REGISTER SHIFTED RIGHT
          0002           EQU       SRL(2)              SHIFT REGISTER SHIFTED LEFT
          0003           EQU       SRLOAD(3)           SHIFT REGISTER LOADED
          PANEL                PAGE   3          DATE: 10/16/75

0072            *        SHIFT INPUT, DATA MUX, AND XMIT MUX ENABLE EQUATES (P4 ACTUALS)
0073      0000           EQU      HIBIT(0)           ZERO FOR H, ONE FOR I
0074      0001           EQU      ENDARND(1)         END AROUND SHIFT
0075      0002           EQU      XMSB(2)            MSB OF X
0076      0003           EQU      BUFFER(3)          SHIFT BUFFER REGISTER

0078      0000           EQU      ENABDMA(0)         DMA ENABLED
0079      0001           EQU      ENABFC(1)          FUNCTION CONTROL REG ENABLED
0080      0002           EQU      ENABMIR(2)         MIR ENABLED
0081      0003           EQU      ENABPMA(3)         BP-MA ENABLED

0083      0000           EQU      HEXGR9(0)          XMIT HEX GREATER THAN 9
0084      0001           EQU      HEXLE9(1)          XMIT HEX LE 9
0085      0002           EQU      ADDR(2)            XMIT ADDRESS FIELD
0086      0003           EQU      XMITCCR(3)         XMIT CONTROL CHAR REGISTER

0088      0000           EQU      COMPUL(0)          COMPLIMENT UPPER/LOWER F/F
0089      0001           EQU      CLRUL(1)           CLEAR UPPER/LOWER F/F
0090      0000           EQU      DIS1ML(0)          FORCE DIS 1. SET MML AT STRBMMX
0091      0001           EQU      DIS1MU(1)          FORCE DIS 1. SET MMU AT STRBMMX
0092      0002           EQU      DISOML(2)          SET MML AT STRBMMX
0093      0003           EQU      DISOMU(3)          SET MMU AT STRBMMX


0095            *        COUNTER ENABLE EQUATES (P5 ACTUALS)
0096      0000           EQU      SAME(0)            COUNTER UNCHANGED
0097      0001           EQU      INCK(1)            COUNTER INCREMENTED BY 1


0099            *        STROBE EQUATES (P6 ACTUALS)
0100      0000           EQU      NULL(00)          NOT USED
0101      0001           EQU      S300(01)          ENABLE S300 TO CPU LSB
0102      0002           EQU      TBRLOAD(02)       LOAD XMIT BUFFER REG
0103      0003           EQU      DRRESET(03)       DATA READY RESET AND SET BUSY
0104      0004           EQU      STRBCCR(04)       STROBE CONTROL CHAR REGISTER
0105      0005           EQU      CLRSR(05)         CLR SHIFT REGISTER
0106      0006           EQU      CLRCT(06)         CLR COUNT, RD STROBE, STROBE BCR
0107      0007           EQU      EXIT(07)          EXIT FROM CPU
0108      0008           EQU      SSREG(08)         STROBE SELECTED REGISTER
0109      000A           EQU      STRBUL(10)        STROBE UPPER/LOWER F/F
0110      000B           EQU      SETERROR(11)      SET ERROR F/F
0111      000C           EQU      STRBMMX(12)       STROBE MM UPPER/LOWER F/F
0112      000D           EQU      CLRBUSY(13)       CLEAR BUSY, CLEAR ERROR
0113      000E           EQU      SETRCPU(14)       CLR TIMING CHAIN (SET RCPU)
0114      000F           EQU      LOADCT(15)        LOAD COUNTER WITH ADDRESS FIELD
```

```
          PANEL                PAGE   4              DATE: 10/16/75

                       *       ADDRESS FIELD EQUATES (P7 ACTUALS)
          0040                 EQU      XHEXGR9($40)        HEX GR 9 AND CONTROL CHAR
          0030                 EQU      XHEXLE9($30)        HEX LE 9
          002A                 EQU      XSTAR($2A)          ASTERISK
          000D                 EQU      XCR($D)             CHARRIAGE RETURN
          000A                 EQU      XLF($A)             LINE FEED
          0020                 EQU      XSPACE($20)         SPACE
```

```
0124                  S          JTRUE      DREADY,,,,,,CHKREM    JUMP IF DATA RDY OR SWITCH DOWN
0124 P0000 0090
0124 P0001 0006
0125                             JTRUE      NOTAUTO,,,,,,S        JUMP IF BUSY OR NOT AUTO-DISPLAY
0125 P0002 01E8
0125 P0003 0000
0126                  OKAY       JFALSE     NOTJ,,,DISOMU,,STRBMMX,JLOC  SET MMU, JUMP IF CC IS J
0126 P0004 0230
0126 P0005 3612
0127                             JFALSE     HORI,,,CLRUL,,STRBUL,KL    JUMP IF NOT H OR I
0127 P0006 0334
0127 P0007 151E
0128                             JFALSE     CTEQZ,,,,,,REPLACE    JUMP IF COUNT NOT EQ 0
0128 P0008 047C
0128 P0009 0015
0129                             JUMP       ,,,,,XMITFCR
0129 P000A 0580
0129 P000B 0011
0130                  CHKREM     JTRUE      REMOTE,,,,,,CHARDY    JUMP IF REMOTE
0130 P000C 0684
0130 P000D 0008
0131                             SEQ        ,,,,TBRLOAD           DEBOUNCE LEADING EDGE
0131 P000E 0780
0131 P000F 0108
0132                  CHARDY     JTRUE      TERM,,,,,,TMRCVD      JUMP IF TERMINATE CODE (COLON)
0132 P0010 08C8
0132 P0011 0010
0133                             JTRUE      ERROR,,,,,,RESET      IGNORE IF ERROR F/F SET
0133 P0012 09BC
0133 P0013 000F
0134                             JTRUE      HEXRCVD,SBLOAD,,,,,ENTERHEX    LD BUFFER, JUMP IF HEX
0134 P0014 0AD3
0134 P0015 000E
0135                             JTRUE      CTEQZ,,,,,,SETCC      SKIP IF COUNT EQUALS ZERO
0135 P0016 0BFC
0135 P0017 000D
0136                  ERRORLOC JUMP         ,,,,SETERROR,RESET    SET ERROR F/F
0136 P0018 0C80
0136 P0019 058F
0137                  SETCC      JUMP       ,,,,STRBCCR,RESET     STRORE CONTRC   AR REGISTER
0137 P001A 0D80
0137 P001B 020F
0138                  ENTERHEX JFALSE       CTEQX3,SBL,SRL,BUFFER,INCK,,*         R TO SHIFT REG
0138 P001C 0E56
0138 P001D B80E
0139                  RESET      JUMP       ,,,,DRRESET,RETURN    RESET AND JU   TO RETURN
0139 P001E 0F80
0139 P001F 01BB
0140                  TMRCVD     JFALSE     ERROR,,,,,DRRESET,OKAY   JUMP I    ERROR
0140 P0020 103C
0140 P0021 0182
0141                  XMITFCR    JUMP       ,SRLOAD,ENABFC,,CLRCT,XMIT32  F      SR, JUMP TO XMIT
0141 P0022 1180
```

```
0141 P0023 D328
0142                JLOC      JFALSE    NOTAUTO,,,,,,XMITFCR  JUMP IF AUTO-DIS. AND NOT BUSY
0142 P0024 1268
0142 P0025 0011
0143                          JTRUE     CTEQZ,,,,,,OTHER16   JUMP IF COUNT EQ 0
0143 P0026 13FC
0143 P0027 001D
0144                          JFALSE    CTLSB,,SRR,CLRUL,INCK,STRBUL,*    POSITION DIGIT NUM
0144 P0028 1458
0144 P0029 5D14
0145                REPLACE   SEQ       ,SRLOAD,ENABFC,,CLRCT  FCR TO SR. STROBE BIT CT REG
0145 P002A 1580
0145 P002B D316
0146                CKEQUAL   JTRUE     EQUAL,,,,,,CKJ       JUMP IF BIT COUNT EQ BIT CT REG
0146 P002C 169C
0146 P002D 0018
0147                          JUMP      ,SRL,ENDARND,INCK,,CKZERO    SHIFT END AROUND
0147 P002E 1780
0147 P002F 981B
0148                CKJ       JTRUE     NOTJ,,,,,,CHGBIT     SKIP IF NOT J
0148 P0030 18B0
0148 P0031 001A
0149                          JUMP      SBL,SRL,BUFFER,INCK,,CKZERO    BUFFER TO SHIFT REG
0149 P0032 1982
0149 P0033 B81B
0150                CHGBIT    SEQ       ,SRL,HIBIT,INCK      CLR OR SET BIT
0150 P0034 1A80
0150 P0035 881B
0151                CKZERO    JFALSE    CTEQ0,,,,,,CKEQUAL   JUMP IF COUNT NOT EQ 0
0151 P0036 1B74
0151 P0037 0016
0152                          JUMP      ,,,,SSREG,XMITFCR    STROBE SR TO FCR
0152 P0038 1C80
0152 P0039 0411
0153                OTHER16   JUMP      ,,COMPUL,,STRBUL,SET16   COMPLIMENT UPPER/LOWER F/F
0153 P003A 1D80
0153 P003B 0527
0154                KL        JFALSE    DIREG,,,,,,NOTDIREG  JUMP IF NOT DIRECT REGISTER
0154 P003C 1E6C
0154 P003D 0040
0155                          JTRUE     CTEQZ,,,,,,CKBPMA    SKIP IF CT EQ 0
0155 P003E 1FFC
0155 P003F 0021
0156                          JFALSE    CTLSB,,,ENABFC,INCK,SSREG,*  STROBE SEL. REG. TWICE
0156 P0040 2058
0156 P0041 1C20
0157                CKBPMA    JFALSE    BPMA,,,,,,CKMIR      SKIP IF NOT BKPT-MA REGISTER
0157 P0042 2128
0157 P0043 0023
0158                          JUMP      ,SRLOAD,ENABPMA,,CLRCT,XMIT32  BKPT-MA TO SR
0158 P0044 2280
0158 P0045 F328
0159                CKMIR     JFALSE    MIR,,,,,,XMITFCR     JUMP IF NOT MIR
```

```
0159 P0046 232C
0159 P0047 0011
0160                    JUMP      ,SRLOAD,ENABMIR,,CLRCT,XMIT32  MIR TO SR
0160 P0048 2480
0160 P0049 E328
0161           XMIT     JFALSE    CPU16,,,,,CLRCT,XMIT32  JUMP IF 32 BIT CPU
0161 P004A 254C
0161 P004B 0328
0162                    JTRUE     MM,,,,,,XMIT32       JUMP IF MM
0162 P004C 26E0
0162 P004D 0028
0163           SET16    SEQ       ,,,,LOADCT,16        SET COUNTER TO 16
0163 P004E 2700
0163 P004F 0790
0164           XMIT32   JTRUE     REMOTE,,,,,,CKFC16   JUMP IF REMOTE
0164 P0050 2884
0164 P0051 002B
0165                    JFALSE    CTEQ31,,SRL,ENDARND,INCK,,*    SHIFT BITS TO LITES
0165 P0052 2940
0165 P0053 9829
0166           TOEND    JUMP      ,,,,CLRBUSY,ENDXMIT  CLEAR BUSY F/F
0166 P0054 2A80
0166 P0055 068A
0167           CKFC15   JTRUE     FC15,,,,,,TOEND      SUPPRESS XMIT IF FC15 SET
0167 P0056 2BA0
0167 P0057 002A
0168                    SEQ       ,,ADDR,,TBRLOAD,XCR  TRANSMIT CHARRIAGE RETURN
0168 P0058 2C00
0168 P0059 210D
0169                    JFALSE    NOTAUTO,,,,,,SKPLF   SKIP LF IF AUTO-DISPLAY
0169 P005A 2D68
0169 P005B 002F
0170                    SEQ       ,,ADDR,,TBRLOAD,XLF  TRANSMIT LINE FEED
0170 P005C 2E00
0170 P005D 210A
0171           SKPLF    JFALSE    ERROR,,,,,,NOERROR   JUMP IF ERROR F/F NOT SET
0171 P005E 2F3C
0171 P005F 0031
0172                    SEQ       ,,ADDR,,TBRLOAD,XSTAR  TRANSMIT ASTERISK
0172 P0060 3000
0172 P0061 212A
0173           NOERROR  SEQ       ,,XMITCCR,,TBRLOAD,XHEXGR9  TRANSMIT CONTROL CHAR
0173 P0062 3100
0173 P0063 3140
0174           HEXDIGIT SEQ       ,,ADDR,,TBRLOAD,XSPACE  TRANSMIT SPACE
0174 P0064 3200
0174 P0065 2120
0175                    JFALSE    LE9,,,,,CLRBUSY,GR9  JUMP IF GREATER THAN 9
0175 P0066 3324
0175 P0067 0636
0176                    SEQ       ,,HEXLE9,,TBRLOAD,XHEXLE9  TRANSMIT HEX DIGIT
0176 P0068 3400
0176 P0069 1130
```

```
0177                           JUMP      .....SHIFT4
0177 P006A 3580
0177 P006B 0037
0178                 GR9       SEQ       ..HEXGR9..TBRLOAD.XHEXGR9  TRANSMIT HEX DIGIT
0178 P006C 3600
0178 P006D 0140
0179                 SHIFT4    JFALSE    CTEQX3..SRL.ENDARND.INCK..*     SHIFT 4 BITS
0179 P006E 3754
0179 P006F 9837
0180                           JFALSE    CTEQ0......HEXDIGIT    SEND AGAIN IF COUNT NOT EQ 0
0180 P0070 3874
0180 P0071 0032
0181                           SEQ       ..ADDR..TBRLOAD.XSPACE  TRANSMIT SPACE
0181 P0072 3900
0181 P0073 2120
0182                 ENDXMIT   SEQ       ....CLRCT             CLEAR COUNTER
0182 P0074 3A80
0182 P0075 0338
0183                 RETURN    JTRUE     REMOTE......S         RETURN IF REMOTE
0183 P0076 3B84
0183 P0077 0000
0184                           JTRUE     DREADY......*         WAIT SWITCH OPEN
0184 P0078 3C90
0184 P0079 003C
0185                           SEQ       ....TBRLOAD           DEBOUNCE TRAILING EDGE
0185 P007A 3D80
0185 P007B 013E
0186                           JTRUE     NOTAUTO.....S          RETURN IF NOT AUTO-DISPLAY
0186 P007C 3EE8
0186 P007D 0000
0187                           JUMP      ..CLRUL..STRRUL.S     CLEAR U/L F/F
0187 P007E 3F80
0187 P007F 1500
0188                 NOTDIREG  JFALSE    RUNNING.....STOPPED JUMP IF STOPPED OR MACRO-MODE
0188 P0080 400C
0188 P0081 0042
0189                           JUMP      ....SETERROR.XMITFCR SET ERROR. JUMP TO XMITFCR
0189 P0082 4180
0189 P0083 0591
0190                 STOPPED   JTRUE     CTEQZ.....SETRCPU.FETCH    JUMP IF COUNT EQ 0
0190 P0084 42FC
0190 P0085 0750
0191                           SEQ       ....CLRCT             CLEAR COUNTER
0191 P0086 4380
0191 P0087 0344
0192                           JFALSE    CTEQ15.LOC0.SRL.XMSB.INCK..*    SHIFT 16 BITS
0192 P0088 4444
0192 P0089 A844
0193                           JFALSE    MM.....CLRCT.NOTMM    JUMP IF NOT MICRO MEMORY
0193 P008A 4560
0193 P008B 0348
0194                           JFALSE    CPU16......NOTMM      SKIP IF NOT 16 BIT CPU
0194 P008C 464C
```

```
0194 P008D 0048
0195                     SEQ        LOC2,,DIS0ML,,STRBMMX      X REG TO MMU, SET MML
0195 P008E 4782
0195 P008F 2648
0196           NOTMM     JFALSE     CTEQ15,LOC0,SRL,XMSB,INCK,,*  SHIFT 16 BITS
0196 P0090 4844
0196 P0091 A848
0197                     JFALSE     MEM,,,,,CLRCT,CKF2    SKIP IF NOT MEM
0197 P0092 495C
0197 P0093 034C
0198                     SEQ        ,,DIS1MU,,STRBMMX     FORCE DISPLAY 1
0198 P0094 4A80
0198 P0095 1648
0199                     SEQ        LOC1,,DIS0MU,,STRBMMX   DIS. 1 TO ADDRESS BUFFER
0199 P0096 4B81
0199 P0097 364C
0200           CKF2      JTRUE      NOTF2,LOC2,,,,,CK32  X REG TO SEL REG, JUMP IF NOT F2
0200 P0098 4CE6
0200 P0099 004E
0201                     SEQ        LOC2                  EXEC LOC2 AGAIN TO WRITE INTO F2
0201 P009A 4D82
0201 P009B 004E
0202           CK32      JFALSE     CPU16,,,,,,SKIPSAVE  JUMP IF 32 BIT CPU
0202 P009C 4E4C
0202 P009D 0053
0203                     JUMP       ,,,,,SAVEX
0203 P009E 4F80
0203 P009F 0052
0204           FETCH     JFALSE     CPU16,,,,,,SAVEX     JUMP IF 32 BIT CPU
0204 P00A0 504C
0204 P00A1 0052
0205                     SEQ        ,,,,LOADCT,16         SET COUNTER TO 16
0205 P00A2 5100
0205 P00A3 0790
0206           SAVEX     JFALSE     CTEQ31,LOC0,SRL,XMSB,INCK,S300,*  SAVE X REGISTER
0206 P00A4 5240
0206 P00A5 A8D2
0207           SKIPSAVE JFALSE      MEM,,,DIS0MU,,STRBMMX,NOTMEM   JUMP IF NOT MEM
0207 P00A6 535C
0207 P00A7 3657
0208                     SEQ        ,,DIS1MU,,STRBMMX     FORCE DISPLAY 1
0208 P00A8 5480
0208 P00A9 1655
0209                     SEQ        LOC1,,DIS0MU,,STRBMMX   DIS. 1 TO ADDRESS BUFFER
0209 P00AA 5581
0209 P00AB 3656
0210                     SEQ        LOC3,,,,CLRCT         READ FROM MEM
0210 P00AC 5683
0210 P00AD 0357
0211           NOTMEM    SEQ        LOC3,,DIS1MU,,STRBMMX  SEL REG TO X, FORCE DISPLAY 1
0211 P00AE 5783
0211 P00AF 1658
0212                     SEQ        0                     REQUIRED PASS BEFORE SHIFT
```

```
           PANEL                PAGE  10               DATE: 10/16/75

0212 P00B0 5880
0212 P00B1 0059
0213                         JFALSE    CTEQ15,LOC0,SRL,XMSB,INCK,S300,*  SHIFT 16 BITS
0213 P00B2 5944
0213 P00B3 A8D9
0214                         JFALSE    CPU16,,,,,CLRCT,NOTCPU16    JUMP IF NOT 16 BIT CPU
0214 P00B4 5A4C
0214 P00B5 035E
0215                         JFALSE    MM,,,DIS1ML,,STRBMMX,NOTCPU16   JUMP IF NOT MM
0215 P00B6 5B60
0215 P00B7 065E
0216                         SEQ       LOC3,,DIS1MU,,STRBMMX  MML TO X, FORCE DISPLAY 1
0216 P00B8 5C83
0216 P00B9 165D
0217                         SEQ       0                         REQUIRED PASS BEFORE SHIFT
0217 P00BA 5D80
0217 P00BB 005E
0218               NOTCPU16 JFALSE    CTEQ15,LOC0,SRL,XMSB,INCK,S300,*  SHIFT 16 BITS
0218 P00BC 5E44
0218 P00BD A8DE
0219                         JUMP      LOC1,,,,EXIT,XMIT    DIS.1 + 1 TO DIS.1, EXIT CPU
0219 P00BE 5F81
0219 P00BF 03A5
0220                         END

  PGM=  00C0 (    192)   COM = 0000 (      0)   DAT = 0000 (      0)
```

```
          PANEL               PAGE  11            DATE: 10/16/75


E Q U I V A L E N C E S
-----------------------

 DEF.LINE  NAME     VALUE              REFFRENCED AT LINE NUMBER

   0000    I        00FF    (000255)
   0026    TRUE     0000    (000000)
   0027    REMOTE   0001    (000001)   0130, 0164, 0183
   0028    SWITCH   0002    (000002)
   0029    RUNNIN   0003    (000003)   0188
   0030    DREADY   0004    (000004)   0124, 0184
   0031    CONDS    0005    (000005)
   0032    EQUAL    0007    (000007)   0146
   0033    FC15     0008    (000008)   0167
   0034    LE9      0009    (000009)   0175
   0035    BPMA     000A    (000010)   0157
   0036    MIR      000B    (000011)   0159
   0037    NOTJ     000C    (000012)   0126, 0148
   0038    HORI     000D    (000013)   0127
   0039    ERROR    000F    (000015)   0133, 0140, 0171
   0040    CTEQ31   0010    (000016)   0165, 0206
   0041    CTEQ15   0011    (000017)   0192, 0196, 0213, 0218
   0042    TERM     0012    (000018)   0132
   0043    CPU16    0013    (000019)   0161, 0194, 0202, 0204, 0214
   0044    HEXRCV   0014    (000020)   0134
   0045    CTEQX3   0015    (000021)   0138, 0179
   0046    CTLSB    0016    (000022)   0144, 0156
   0047    MEM      0017    (000023)   0197, 0207
   0048    MM       0018    (000024)   0162, 0193, 0215
   0049    NOTF2    0019    (000025)   0200
   0050    NOTAUT   001A    (000026)   0125, 0142, 0169, 0186
   0051    DIREG    001B    (000027)   0154
   0052    CTEQ0    001D    (000029)   0151, 0180
   0053    CTEQZ    001F    (000031)   0128, 0135, 0143, 0155, 0190
   0056    LOC0     0000    (000000)   0192, 0196, 0206, 0213, 0218
   0057    LOC1     0001    (000001)   0199, 0209, 0219
   0058    LOC2     0002    (000002)   0195, 0200, 0201
   0059    LOC3     0003    (000003)   0210, 0211, 0216
   0061    SBNOP    0000    (000000)
   0062    SBR      0001    (000001)
   0063    SBL      0002    (000002)   0138, 0149
   0064    SBLOAD   0003    (000003)   0134
   0067    SRNOP    0000    (000000)
```

```
          PANEL                 PAGE  12              DATE: 10/16/75

0068     SRR     0001    (000001)    0144
0069     SRL     0002    (000002)    0138, 0147, 0149, 0150, 0165, 0179, 0192, 0196, 0206, 0213, 0218
0070     SRLOAD  0003    (000003)    0141, 0145, 0158, 0160
0073     HIBIT   0000    (000000)    0150
0074     ENDARN  0001    (000001)    0147, 0165, 0179
0075     XMSB    0002    (000002)    0192, 0196, 0206, 0213, 0218
0076     BUFFER  0003    (000003)    0138, 0149
0078     ENABDM  0000    (000000)
0079     ENABFC  0001    (000001)    0141, 0145, 0156
0080     ENABMI  0002    (000002)    0160
0081     ENABPM  0003    (000003)    0158
0083     HEXGR9  0000    (000000)    0178
0084     HEXLE9  0001    (000001)    0176
0085     ADDR    0002    (000002)    0168, 0170, 0172, 0174, 0181
0086     XMITCC  0003    (000003)    0173
0088     COMPUL  0000    (000000)    0153
0089     CLRUL   0001    (000001)    0127, 0144, 0187
0090     DIS1ML  0000    (000000)    0215
0091     DIS1MU  0001    (000001)    0198, 0208, 0211, 0216
0092     DIS0ML  0002    (000002)    0195
0093     DIS0MU  0003    (000003)    0126, 0199, 0207, 0209
0096     SAME    0000    (000000)
0097     INCK    0001    (000001)    0138, 0144, 0147, 0149, 0150, 0156, 0165, 0179, 0192, 0196, 0206, 0213, 0218
0100     NULL    0000    (000000)
0101     S300    0001    (000001)    0206, 0213, 0218
0102     TBRLOA  0002    (000002)    0131, 0168, 0170, 0172, 0173, 0174, 0176, 0178, 0181, 0185
0103     DRRESE  0003    (000003)    0139, 0140
0104     STRBCC  0004    (000004)    0137
0105     CLRSR   0005    (000005)
0106     CLRCT   0006    (000006)    0141, 0145, 0158, 0160, 0161, 0182, 0191, 0193, 0197, 0210, 0214
0107     EXIT    0007    (000007)    0219
0108     SSREG   0008    (000008)    0152, 0156
0109     STRBUL  000A    (000010)    0127, 0144, 0153, 0187
0110     SETERR  000B    (000011)    0136, 0189
0111     STRBMM  000C    (000012)    0126, 0195, 0198, 0199, 0207, 0208, 0209, 0211, 0215, 0216
0112     CLRBUS  000D    (000013)    0166, 0175
0113     SETRCP  000E    (000014)    0190
0114     LOADCT  000F    (000015)    0163, 0205
0117     XHEXGR  0040    (000064)    0173, 0178
0118     XHEXLE  0030    (000048)    0176
0119     XSTAR   002A    (000042)    0172
0120     XCR     000D    (000013)    0168
0121     XLF     000A    (000010)    0170
0122     XSPACE  0020    (000032)    0174, 0181
```

```
S Y M B O L S
-------------

DEF.LINE NAME    ADDRESS           REFERENCED AT LINE NUMBER
  0124   S       0000                0124, 0124, 0125, 0125, 0125, 0126, 0126, 0127, 0127, 0128, 0128, 0129, 0129, 0130, 0130, 0131
                                     0131, 0132, 0132, 0133, 0133, 0134, 0134, 0135, 0135, 0136, 0136, 0137, 0137, 0138, 0138, 0139
                                     0139, 0140, 0140, 0141, 0141, 0142, 0142, 0143, 0143, 0144, 0144, 0145, 0145, 0146, 0146, 0147
                                     0147, 0148, 0148, 0149, 0149, 0150, 0150, 0151, 0151, 0152, 0152, 0153, 0153, 0154, 0154, 0155
                                     0155, 0156, 0156, 0157, 0157, 0158, 0158, 0159, 0159, 0160, 0160, 0161, 0161, 0162, 0162, 0163
                                     0164, 0164, 0165, 0165, 0166, 0166, 0167, 0167, 0168, 0169, 0169, 0170, 0171, 0171, 0172, 0173
                                     0174, 0175, 0175, 0176, 0177, 0177, 0178, 0179, 0179, 0180, 0180, 0181, 0182, 0182, 0183, 0183
                                     0183, 0184, 0184, 0185, 0185, 0186, 0186, 0186, 0187, 0187, 0187, 0188, 0188, 0189, 0189, 0190
                                     0190, 0191, 0191, 0192, 0192, 0193, 0193, 0194, 0194, 0195, 0195, 0196, 0196, 0197, 0197, 0198
                                     0198, 0199, 0199, 0200, 0200, 0201, 0201, 0202, 0202, 0203, 0203, 0204, 0204, 0205, 0206, 0206
                                     0207, 0207, 0208, 0208, 0209, 0209, 0210, 0210, 0211, 0211, 0212, 0212, 0213, 0213, 0214, 0214
                                     0215, 0215, 0216, 0216, 0217, 0217, 0218, 0218, 0219, 0219
  0126   OKAY    0004                0140
  0130   CHKREM  000C                0124
  0132   CHARDY  0010                0130
  0136   ERRORL  0018
  0137   SETCC   001A                0135
  0138   ENTERH  001C                0134
  0139   RESET   001E                0133, 0136, 0137
  0140   TMRCVD  0020                0132
  0141   XMITFC  0022                0129, 0142, 0152, 0159, 0189
  0142   JLOC    0024                0126
  0145   REPLAC  002A                0128
  0146   CKEQUA  002C                0151
  0148   CKJ     0030                0146
  0150   CHGBIT  0034                0148
  0151   CKZERO  0036                0147, 0149
  0153   OTHERI  003A                0143
  0154   KL      003C                0127
  0157   CKBPMA  0042                0155
  0159   CKMIR   0046                0157
  0161   XMIT    004A                0219
  0163   SET16   004E                0153
  0164   XMIT32  0050                0141, 0158, 0160, 0161, 0162
  0166   TOEND   0054                0167
  0167   CKFC15  0056                0164
  0171   SKPLF   005E                0169
  0173   NOERRO  0062                0171
  0174   HEXDIG  0064                0180
  0178   GR9     006C                0175
  0179   SHIFT4  006E                0177
```

```
0182    ENDXMI  0074            0166
0183    RETURN  0076            0139
0188    NOTDIR  0080            0154
0190    STOPPE  0084            0188
0196    NOTMM   0090            0193, 0194
0200    CKF2    0098            0197
0202    CK32    009C            0200
0204    FETCH   00A0            0190
0206    SAVEX   00A4            0203, 0204
0207    SKIPSA  00A6            0202
0211    NOTMEM  00AE            0207
0218    NOTCPU  00BC            0214, 0215
```

```
          PANEL                 PAGE  15              DATE: 10/16/75

*** A L P H A B E T I C A L   S O R T   O F   S Y M B O L S ***

 ADDR    0085  BPMA    0035  BUFFER  0076  CHARDY  0132  CHGBIT  0150  CHKPEM  0130  CK32    0202  CKBPMA  0157  CKEQUA  0146
 CKF2    0200  CKFC15  0167  CKJ     0148  CKMIR   0159  CKZERO  0151  CLRBUS  0112  CLRCT   0106  CLRSR   0105  CLRUL   0089
 COMPUL  0088  CONDS   0031  CPU16   0043  CTEQ0   0052  CTEQ15  0041  CTEQ31  0040  CTEQX3  0045  CTEQZ   0053  CTLSB   0046
 DIREG   0051  DISOML  0092  DISOMU  0093  DISIML  0090  DISIMU  0091  DREADY  0030  DRRESE  0103  ENABDM  0078  ENABFC  0079
 ENABMI  0080  ENABPM  0081  ENDARN  0074  ENDXMI  0182  ENTERH  0138  EQUAL   0032  ERROR   0039  ERRORL  0136  EXIT    0107
 FC15    0033  FETCH   0204  GR9     0178  HEXDIG  0174  HEXGR9  0083  HEXLF9  0084  HEXRCV  0044  HIBIT   0073  HORI    0038
 I       0000  INCK    0097  JLOC    0142  KL      0154  LF9     0034  LOADCT  0114  LOC0    0056  LOC1    0057  LOC2    0058
 LOC3    0059  MEM     0047  MIR     0036  MM      0048  NOERRO  0173  NOTAUT  0050  NOTCPU  0218  NOTDIR  0188  NOTF2   0049
 NOTJ    0037  NOTMEM  0211  NOTMM   0196  NULL    0100  OKAY    0126  OTHER1  0153  REMOTE  0027  REPLAC  0145  RESET   0139
 RETURN  0183  RUNNIN  0029  S       0124  S300    0101  SAME    0096  SAVEX   0206  SBL     0063  SBLOAD  0064  SBNOP   0061
 SBR     0062  SET15   0163  SETCC   0137  SETERR  0110  SETRCP  0113  SHIFT4  0179  SKIPSA  0207  SKPLF   0171  SRL     0069
 SRLOAD  0070  SRNOP   0067  SRR     0068  SSREG   0108  STOPPE  0190  STRBCC  0104  STRBMM  0111  STRBUL  0109  SWITCH  0028
 TBRLOA  0102  TERM    0042  TMRCVD  0140  TOEND   0165  TRUE    0026  XCR     0120  XHEXGR  0117  XHEXLE  0118  XLF     0121
 XMIT    0161  XMIT32  0164  XMITCC  0086  XMITFC  0141  XMSB    0075  XSPACE  0122  XSTAR   0119
```

UART DETAILS

This appendix contains the detailed descriptions of the operating characteristics applicable to the UART terminals (table B-1), a functional block diagram of the UART (figure B-1), and the controlware program listing of the breakpoint controller. The UART provides the RS232 communication interface between the breakpoint controller and RS232 compatible peripherals. The controlware program sequences and controls the breakpoint controller operations. For a detailed description of the controlware format, refer to the paragraph on Breakpoint Controller Controlware in section 4.

TABLE B-1. UART PIN DESIGNATIONS AND DESCRIPTIONS

Designation	Pin No.	Description	Function
RRD (RBRD)	4	Receiver register disconnect	A high-level input voltage, V_{IH}, applied to this line disconnects the receiver holding register outputs from the RR8-RR1 data outputs (pins 5-12).
RR8-RR1	5-12	Receiver buffer (holding) register data	The parallel contents of the receiver register appear on these lines if a low-level input voltage, V_{IL}, is applied to RRD. For character formats of fewer than eight bits, received characters are right-justified (RR1 = LSB) and the truncated bits are forced to a low output voltage V_{OL}.
RPE	13	Parity error	A high-level output voltage, V_{OH}, on this line indicates that the received parity does not compare to that programmed by the even parity enable (PS) control line (pin 39). This output is updated each time a character is transferred to the receiver buffer register. RPE lines from a number of arrays can be bussed together since an output disconnect capability is provided by the status flag disconnect (SFD) line (pin 16).
RFE	14	Framing error	A high-level output voltage, V_{OH}, on this line indicates that the received character has no valid stop bit; that is, the bit following the parity bit (if programmed) is not a high-level voltage. This output is updated each time a character is transferred to the receiver holding register. RFE lines from a number of arrays can be bussed together, since an output disconnect capability is provided by the status flag disconnect (SFD) line (pin 16).
ROE	15	Over-run error	A high-level output voltage, V_{OH}, on this line indicates that the data received flag RDR (pin 19) was not reset before the next character was transferred to the receiver holding register. ROE lines from a number of arrays can be bussed together, since an output disconnect capability is provided by the status flag disconnect (SFD) line (pin 16).
SFD	16	Status flag disconnect	A high-level input voltage, V_{IH}, applied to this pin disconnects RPE, RFE, ROE, RDR, and TBRE, allowing them to be bus-connected.
RRC	17	Receiver register	The receiver clock frequency is 16 times the desired receiver shift rate.
DRR	18	Data received reset	A low-level input voltage, V_{IL}, applied to this line resets the DRR line.
RDR	19	Data ready	A high-level output voltage, V_{OH}, indicates that an entire character has been received and transferred to the receiver holding register.
RRI	20	Receiver input	Serial input data received on this line enters the receiver register at a point determined by the character length, parity, and the number of stop bits. A high-level input voltage, V_{IH}, must be present when data is not being received.

TABLE B-1. UART PIN DESIGNATIONS AND DESCRIPTIONS (Contd)

Designation	Pin No.	Description	Function
MR	21	Master reset	This line is strobed to a high-level input voltage, V_{IH}, to clear the logic. It resets the transmitter and receiver registers, the receiver holding register, RFE, ROE, RPE, and DRR, and sets the serial output line to a high-level output voltage, V_{OH}.
TBRE	22	Transmitter buffer register empty	A high-level output voltage, V_{OH}, on this line indicates that the transmitter holding register has transferred its contents to the transmitter register and may be loaded with a new character.
TBRL	23	Transmitter buffer register load	A low-level input voltage, V_{IL}, applied to this line enters a character into the transmitter holding register. A transition from a low-level input voltage, V_{IL}, to a high-level input voltage, V_{IH}, transfers the character into the transmitter register if the register is not in the process of transmitting a character. If a character is being transmitted, the transfer is delayed until the transmission is completed. Upon completion, the new character is automatically transferred simultaneously with the initiation of the serial transmission of the new character.
TRE	24	Transmitter register empty	A high-level output voltage, V_{OH}, on this line indicates that the transmitter register has completed serial transmission of a full character including stop bits. It remains at this level until the start of transmission of the next character.
TRO	25	Transmitter register output	The contents of the transmitter register (start bit, data bits, parity bit, and stop (bit), are serially shifted out on this line. When no data is being transmitted, this line remains at a high-level output voltage, V_{OH}. Start of transmission is defined as the transition of the start bit from a high-level output voltage, V_{OH}, to a low-level voltage, V_{OL}.
TR1-TR8	26-33	Transmitter register, data inputs	The character to be transmitted is loaded into the transmitter holding register on these lines with the TBRL strobe. If a character of less than 8 bits has been selected (by WLS1 and WLS2), the character is right-justified to the least significant bit, RR1, and the excess bits are disregarded. A high-level input voltage, V_{IH}, causes a transmission of a high-level output voltage, V_{OH}.
CRL	34	Control register load	A high-level input voltage, V_{IH}, on this line loads the control register with the control bits (WLS1, WLS2, RPE, PI, and SBS). This line may be strobed or hard-wired to a high-level input voltage, V_{IH}.
PI	35	Parity inhibit	A high-level input voltage, V_{IH}, on this line inhibits the parity generation and verification circuits and clamps the RPE output (pin 13) to V_{OL}. If parity is inhibited, the stop bits immediately follow the last data bit on transmission.
SBS	36	Stop bit(s) select	This line selects the number of stop bits to be transmitted after the parity bit. A high-level input voltage, V_{IH}, on this line selects two stop bits; a low-level input voltage, V_{IL}, selects a single stop bit. Selection of two stop bits when programming a 5-bit word generates 1/5 bits from the UART.
WLS2-WLS1	37,38	Word length select	These lines select the character length (exclusive of parity) as follows: WLS2 / WLS1 / Word Length 0 / 0 / 5 bits 0 / 1 / 6 bits 1 / 0 / 7 bits 1 / 1 / 8 bits
PS	39	Parity select	This line determines whether even or odd parity is to be generated by the transmitter and checked by the receiver. A high-level input voltage, V_{IH}, selects even parity; a low-level input voltage, V_{IL}, selects odd parity.
TRC	40	Transmitter register clock	The transmitter clock frequency is 16 times the desired transmitter rate.

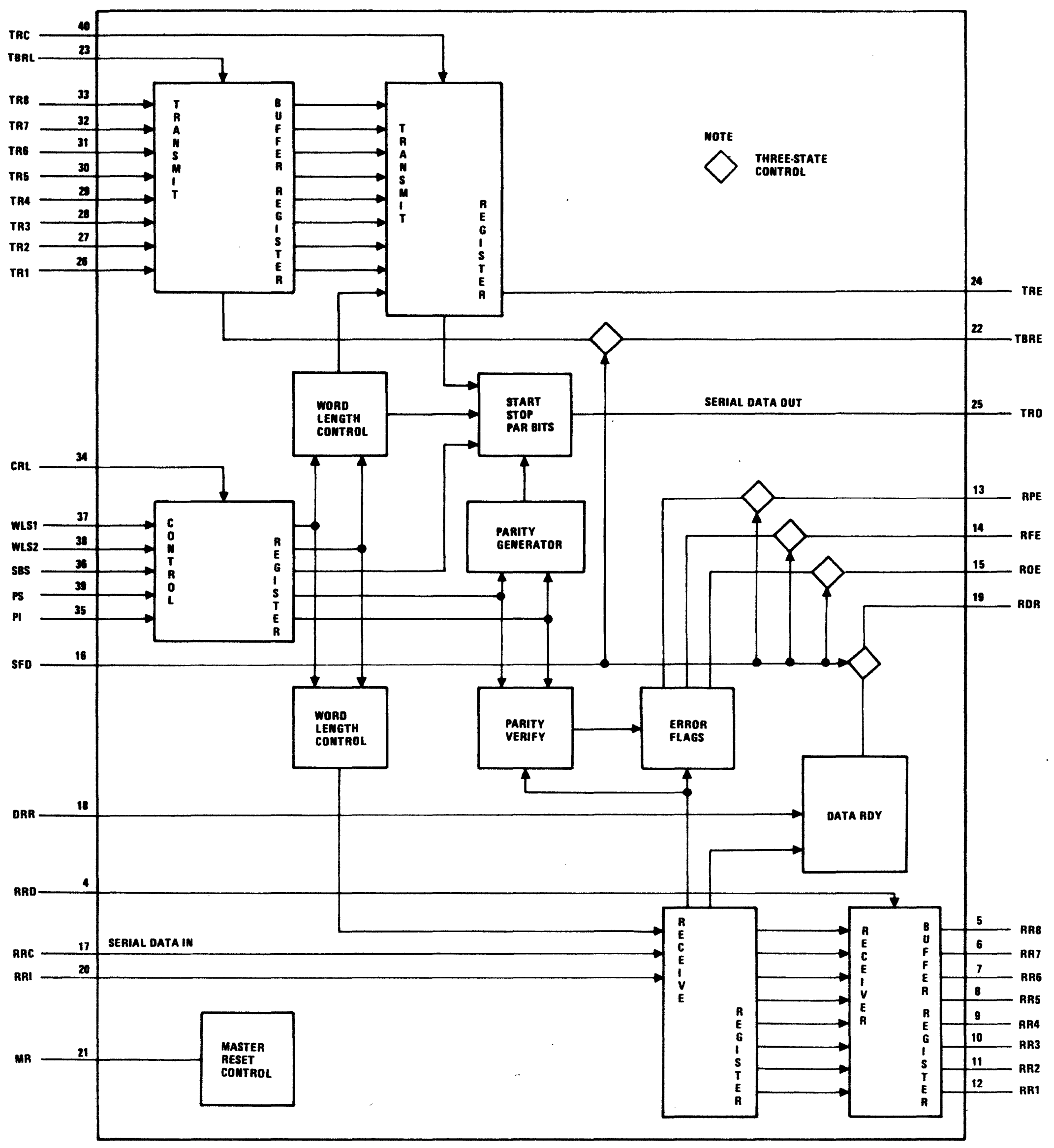


Figure B-1. UART Functional Block Diagram

COMMENT SHEET

MANUAL TITLE DT195-A (FC402-A, DT120-A) Breakpoint Controller and Breakpoint Panel

Hardware Maintenance Manual

PUBLICATION NO. 96729000 REVISION A

FROM NAME:

BUSINESS
ADDRESS:

COMMENTS: This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number.

CUT ALONG LINE

STAPLE STAPLE

FOLD

FIRST CLASS
PERMIT NO. 333

LA JOLLA, CA.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
PUBLICATIONS AND GRAPHICS DIVISION
4455 EASTGATE MALL
LA JOLLA, CALIFORNIA 92037

CUT ALONG LINE

FOLD

STAPLE STAPLE