

.REM 8

IDENTIFICATION
.....

PRODUCT CODE: AC-E648F-MC
PRODUCT NAME: CZTUUF0 TU58 PERF EXER
PRODUCT DATE: 23 JANUARY 1984
MAINTAINER: TAPE DIAGNOSTIC ENGINEERING
AUTHOR: R. J. ROSS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1979,1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL PDP UNICUS MASSBUS
DEC DECUS DECTAPE

HISTORY

JUNE 18, 1979	INITIAL RELEASE	CZTUJAO
JULY 1, 1979	SECOND RELEASE	CZTUJBO
JUNE 1, 1980	THIRD RELEASE	CZTUJBI
OCTOBER 1, 1981	FOURTH RELEASE	CZTUJCO
MARCH 1, 1982	FIFTH RELEASE	CZTUJDO
JUNE 1, 1983	SIXTH RELEASE	CZTUJEO
JANUARY 23, 1984	SEVENTH RELEASE	CZTUJFO

CZTUJAO

1. INITIAL RELEASE -- PERF. EXER. FOR UP TO 8 TU58 CONTROLLERS WITH ONE OR TWO DRIVES EACH.

CHANGES TO CZTUJAO

1. THE PROGRAM WAS MODIFIED TO RUN UNDER THE NEW DIAGNOSTIC SUPERVISOR CHSAO. AS A RESULT OF THIS CONVERSION, THIS PROGRAM NOW OPERATES IN 8K AND PAPER TAPE DISTRIBUTION REQUIRES ONLY ONE PART AK-E650B-MC.

CHANGES TO CZTUJBO

1. "CLR @ XMSR(R5)" HAS BEEN CHANGED TO "DEC @ XMSR(R5)" TO ALLEVIATE THE PROBLEM OF DESTROYING ANY PREVIOUSLY SET PROGRAMMABLE SPEED IN THE DLV11-E,F, OR DC319 DLART WHEN THE TU58 INIT SEQUENCE WAS TERMINATED.

CHANGES TO CZTUJBI

1. TEST 9 WAS ADDED TO THE DIAGNOSTICS BECAUSE THE TU58 HAS BEEN UPDATED TO USE MODIFIED RADIAL SERIAL PROTOCOL.

CHANGES TO CZTUJCO

1. A TEST WAS ADDED TO VERIFY 128 BYTE/BLOCK MODE. THE TEST IS SIMILAR TO TEST 3. IT WRITES, READS, AND VERIFIES SEQUENTIAL BLOCKS OF TAPE FROM BLOCK 0 THROUGH BLOCK 2047. THIS IS DONE FOR EACH SELECTED DRIVE IN EACH SELECTED UNIT. THIS WILL BE TEST 4. TEST NUMBERED 4-8 WILL BECOME TEST 5-9.
2. IN TEST 9, 'MRSP' WILL BE TESTED DIFFERENTLY. IN THIS VERSION TO TEST THE NEED FOR HANDSHAKING. THE WAIT LOOP IS BEFORE SENDING THE 'CONTINUE' INSTEAD OF AFTER. THIS WILL VERIFY THAT THE TU58 CANNOT SEND DATA WITHOUT A HANDSHAKE.

D1

USER DOCUMENTATION MACY11 30(1046) 25-JAN-84 08:33 PAGE 7-1
CZTUUF.P11 25-JAN-84 08:09 M%CNTOPI GPRM COUNT OPTION

SEQ 0003

CHANGES TO CZTUUEO

1. ADDED SOFTWARE PARAMETER TO ALLOW OPTION OF EXECUTING TEST 3 ON DRIVE 0 ONLY, OR ALL DRIVES. IF TEST 3 IS EXECUTED ON DRIVE 0 ONLY, EXECUTION TIME IS REDUCED.

E2

GLOBAL AREAS MACY11 30(1046) 25-JAN-84 08:33 PAGE 9-7
CZTUUF.P11 25-JAN-84 08:09 GLOBAL EQUATES SECTION

SEQ 0017

(1)	000300	PRI06-- 300
(1)	000240	PRI05-- 240
(1)	000200	PRI04-- 200
(1)	000140	PRI03-- 140
(1)	000100	PRI02-- 100
(1)	000040	PRI01-- 40
(1)	000000	PRI00-- 0
(1)		;
(1)		;OPERATOR FLAG BITS
(1)		;
(1)	000004	EVL-- 4
(1)	000010	LOT-- 10
(1)	000020	ADR-- 20
(1)	000040	IDU-- 40
(1)	000100	ISR-- 100
(1)	000200	UAM-- 200
(1)	000400	BOE-- 400
(1)	001000	PNT-- 1000
(1)	002000	PRI-- 2000
(1)	004000	IXE-- 4000
(1)	010000	IBE-- 10000
(1)	020000	IER-- 20000
(1)	040000	LOE-- 40000
(1)	100000	HOE-- 100000

3963

```

3976
3977
3978
3979
3980
3981
3982      000002      SFTRD      ==      2
3983      000004      SFTWR      ==      4
3984      000006      RCINIT     ==      6
3985      000010      OTL        ==      8.
3986      000012      OVRN       ==     10.
3987      000014      BDCOM      ==     12.
3988      000016      HRDRD     ==     14.
3989      000020      HRDWR     ==     16.
3990      000022      BDCHK      ==     18.
3991      000024      SKERR      ==     20.
3992      000026      WRLOCK    ==     22.
3993      000030      NOMOT     ==     24.
3994      000032      CNINIT    ==     26.
3995      000034      PARTL     ==     28.
3996      000036      NOUNIT    ==     30.
3997      000040      CMNDR     ==     32.
3998      000042      RECERR    ==     34.
3999      000044      SLFER     ==     36.
4000      000046      SUCOTL    ==     38.
4001      000050      TORCVB    ==     40.
4002      000054      NCART     ==     44.
4003      000056      TOSNDB    ==     46.
4004
4005      ;           IN ADDITION, SYSTEM SETUP OR RUNTIME ERRORS ARE:
4006
4007      ;           100. - ALL UNITS ABORTED
4008
4009      ;           101. - MORE THAN 8. UNITS (16 DRIVES) REQUESTED
4010
4011      ;           102. - NEITHER DRIVE SELECTED FOR THIS CONTROLLER
4012
4013      ;           ALL THE ABOVE ARE CLASSIFIED AS SYSTEM FATAL
4014

```

```

4016      ,SBTTL  GENERAL EQUATES
4017      ;RADIAL SERIAL CODES:
4018      ;-----
4019      ;THE FLAG BYTE CODES ARE:
4020      000002      RSCMND  == 2           ;"COMMAND" PACKET
4021      000020      RSCONT  == 20          ;"CONTINUE" SINGLE BYTE
4022      000020      RSXON   == 20          ;"XON" SINGLE BYTE
4023      000023      RSXOFF  == 23          ;"XOFF" SINGLE BYTE
4024      000004      RSNIT   == 4           ;"INIT" SINGLE BYTE
4025      000001      RSDATA  == 1           ;"DATA" PACKET
4026      000002      RSEND   == RSCMND     ;"END" PACKET FLAG IS "COMMAND"
4027      ;-----
4028      ;END PACK SIZE:
4029      000016      RSNDSZ  == 14.         ;TOTAL BYTES IN COMMAND PACKET
4030      ;MESSAGE PACK SIZE:
4031      000012      RMSIZ   == 12          ;10. BYTES FOR BYTE COUNT INSIDE CMND PACK
4032      ;DATA PACK SIZE:
4033      000204      RSDASZ  == 132.        ;TOTAL BYTES IN DATA PACKET
4034      ;DATA + END PACK SIZE:
4035      000222      RSDNSZ  == RSDASZ+RSNDSZ
4036      ;GET CHARACTERISTICS DATA PACKET SIZE
4037      000034      RSGCDP  == 28.         ;TOTAL BYTES FOR GET CHAR DATA PACKET
4038      ;MINUS THE END PACKET
4039      000016      RSSNSZ  == RSMSIZ + 4   ;SIZE FOR SENDING COMMAND PACK
4040      001036      RCBFSZ  == 4*RSDASZ+RSNDSZ ;4 DATA PAKS AND END PACK
4041      ;IS SIZE OF RCV BUFFERS
4042      ;-----
4043      ;
4044      ;THE OP CODES ARE:
4045      000100      RSSEND  == 100          ;END PACK DESCRIPTOR
4046      000003      RSSWR   == 3           ;WRITE
4047      000002      RSSRD   == 2           ;READ
4048      000005      RSSSEK  == 5           ;SEEK
4049      000012      RSSGET  == 12          ;GET CHARACTERISTICS
4050      000000      RSSNOP  == 0           ;NO-OPERATION
4051      000001      RSSNIT  == 1           ;INITIALIZE
4052      000007      RSSSLF  == 7           ;SELF TEST
4053      ;-----
4054      ;THE SUCCESS CODES ARE:
4055      177720      ESABO   ==-48.         ;BAD COMMAND FROM HOST
4056      177767      ESNCR   ==-9.         ;NO CARTRIDGE
4057      177770      ESNONX  ==-8.         ;NO DRIVE
4058      000000      ESOK    ==0           ;OP COMPLETE SUCCESS
4059      177776      ESPART  ==-2          ;PARTIAL OP
4060      177740      ESSK    ==-32.        ;SEEK ERROR
4061      000001      ESTRY   ==1           ;RETRY OCCURRED
4062      177765      ESWLOC  ==-11.        ;WRITE PROTECTED
4063      177737      ESNOMO  ==-33.        ;MOTOR STOPPED
4064      177720      ESCMD   ==-48.        ;COMMAND ERROR
4065      177711      ESREC   ==-55.        ;BAD RECORD NUMBER
4066      177757      ESCK   ==-17.        ;TU CHKSUM ERROR
4067      177777      ESSLF   ==-1.         ;SELF TEST ERROR
4068      177757      ESCKSM  =ESCK
4069      177757      ESWR    =ESCK
4070      177757      ESRD    =ESCK
;-----

```

4072			.SBTTL ERROR MESSAGE DESCRIPTIONS
4073			
4074			;THE TABLE OF ERROR MESSAGES (ADDRESSES). ABNDX(R5) CONTAINS THE OFFSET
4075			;OF THE REASON. IT'S ABSOLUTE ADDRESS IS RSNTAB + ABNDX(R5).
4076			
4077	002230	002324	RSNTAB: MSNLOG
4078	002232	003056	MSSFRD
4079	002234	003116	MSSFWR
4080	002236	002540	MSRNIT
4081	002240	003002	MSQRSP
4082	002242	003262	MSOVRN
4083	002244	002406	MSCOM
4084	002246	003156	MSHORD
4085	002250	003220	MSHOWR
4086	002252	002560	MSHCHK
4087	002254	002310	MSSKER
4088	002256	002516	MSWPRO
4089	002260	002450	MSNOMO
4090	002262	002622	MSNIT
4091	002264	002636	MSPART
4092	002266	002660	MSUNIT
4093	002270	002706	MSCMD
4094	002272	002722	MSREC
4095	002274	002366	MSELF
4096	002276	002742	MSWRSP
4097	002300	002766	MSNRSP
4098	002302	002324	MSNLOG
4099	002304	002466	MSNOTP
4100	002306	003034	MSTOSN

5.0 DEVICE INFORMATION TABLES

CONSULT SECTION SUBTITLED "DATA BLOCK FORMAT" FURTHER ON IN THIS LISTING.

6.0 TEST SUMMARIES

INIT: INIT IS SENT TO DEVICE IF:
OR
1. INIT CODE IN SUPERVISOR IS EXECUTED
2. INIT IS REQUESTED BY DEVICE AS A RESULT OF ERROR.

TEST 1: INITIATES FIRMWARE DIAGNOSTICS AT DEVICE LEVEL (SELF TEST)

TEST 2: SEEK TEST. SEEKS BOT ON BOTH TRACKS, THEN VERIFIES 60 IPS OPERATION TO SEEK EOT ON BOTH TRACKS, ENDING THEN AT BOT.

TEST 3: PERFORMS WRITE, THEN READ OF ADJACENT BLOCKS AT BOT WITH VARYING DATA, THEN SEEKS HALF WAY INTO REMAINING TAPE AND REPEATS THE ABOVE UNTIL EOT. THIS TEST IS IN 512 BYTE/BLOCK MODE.

TEST 4: PERFORMS WRITE, THEN READ OF ADJACENT BLOCKS AT BOT WITH VARYING DATA, THEN SEEKS HALF WAY INTO REMAINING TAPE AND REPEATS THE ABOVE UNTIL EOT. THIS TEST IS IN 128 BYTE/BLOCK MODE.

TESTS 5-8: READS OR WRITES BLOCK # AS DATA INTO SUCCESSIVE BLOCKS ON TAPE, THE LENGTH OF WHICH IS DETERMINED BY SOFTWARE QUESTION #1: DEFAULT IS SHORT TAPE (8.) MINIMUM, (8.) RESULTS IN TRANSFER OF 3. (OR 4 PER TRACK) 512. BYTE BLOCKS OF DATA PER READ (OR WRITE) OPERATION. THE ALGORITHM SWITCHES TRACKS REGARDLESS OF THE NUMBER BLOCKS SELECTED. DRIVE NUMBER IS ADDED TO RECORD AS DEFAULT, SO FOR TAPE INTERCHANGE TESTING. ANSWER (N) TO SOFTWARE (SW) QUESTION #2.

NOTE: THE AMOUNT OF TIME SPENT IN TESTS 5-8 IS QUITE LONG IF THE FULL TAPE (512.) IS SELECTED.

TEST 5: WRITE TAPE
TEST 6: READ TAPE
TEST 7: 'WRITE VERIFY' TAPE
TEST 8: READ MODIFIED THRESHOLD TAPE

TEST 9: THE FIRST PART OF TEST 9 DETERMINES IF A UNIT IS CAPABLE OF MODIFIED RADIAL SERIAL PROTOCOL. THIS PART OF THE TEST IS WRITTEN USING RADIAL SERIAL PROTOCOL, AND DETERMINES THE PROTOCOL OF A UNIT BY SENDING THE TUS8 A GET CHARACTERISTICS COMMAND AND MONITORING THE RESPONSE. IF THE TUS8 RETURNS AN END PACKET IT IS A MODIFIED UNIT. IF THE TUS8 RETURNS A DATA PACKET IT IS A NON-MODIFIED UNIT. NOTE, THE DATA PACKET RETURNED ON A GET CHARAC RISTICS COMMAND IS NOT NORMAL, RATHER IT CONSISTS OF A DATA PACKET THAT IS 28. BYTES PLUS AN END PACKET WHICH IS 14. BYTES. THE SECOND PART OF TEST 9 TESTS ONLY THOUGH5 UNITS THAT ARE MODIFIED. THIS IS ACHIEVED BY LETTING NON-MODIFIED UNITS JUMP OVER CODE. IT WAS ASSUMED THAT IF A UNIT CAN READ,WRITE,ETC... WHEN OPERATING IN RSP, THEN IT CAN READ,WRITE,ETC... WHEN OPERATING IN MRSP. THEREFORE ALL THAT HAD TO BE TESTED WAS THE ABILITY OF MODIFIED UNIT TO BE ABLE TO SEND ONE BYTE AND WAIT FOR A CONTINUE FROM THE HOST BEFORE SENDING THE NEXT BYTE. A PROTOCOL SUMMARY OF THE UNITS IS ADVAIABLE BY ANSWERING YES (Y) TO SOFTWARE (SW) QUESTION # 5.

3765
3766
3792
3794
3795 002000
3797
3798 002000
3799
3800
3801
3802
3803
3804
3805 002000
3806
3814
3815 002000
(4) 002000
(4) 002000 103
(4) 002001 132
(4) 002002 124
(4) 002003 125
(4) 002004 125
(6) 002005 000
(6) 002006 000
(5) 002007 000
(5) 002010
(4) 002010 106
(5) 002011
(4) 002011 060
(5) 002012
(4) 002012 000 01
(5) 002014
(4) 002014 007020
(5) 002016
(4) 002016 041366
(5) 002020
(4) 002020 041530
(5) 002022
(4) 002022 002176
(5) 002024
(4) 002024 002210
(5) 002026
(4) 002026 042236
(5) 002030
(4) 002030 000000
(5) 002032
(4) 002032 000000
(5) 002034
(4) 002034 000001
(5) 002036
(4) 002036 000000
(5) 002040
(4) 002040 002152
(5) 002042
(4) 002042 000340

.TITLE PROGRAM HEADER AND TABLES
.SBTTL PROGRAM HEADER
.ENABL ABS,AMA 2000
.NLIST BEX
BGNMOD

; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
!--

POINTER BGNRPT,BGNSW,BGNSFT,BGNRU,BGNDU,BGNSETUP

HEADER CZTUUF,0,3600,,1,PRI07

L\$NAME::
.ASCII /C/
.ASCII /Z/
.ASCII /I/
.ASCII /U/
.ASCII /U/
.BYTE 0
.BYTE 0
.BYTE 0
L\$REV::
.ASCII /F/
L\$DEPO::
.ASCII /O/
L\$UNIT::
.WORD T\$PTHV
L\$TIML::
.WORD 3600.
L\$HPCP::
.WORD L\$HARD
L\$SPCP::
.WORD L\$SOFT
L\$HPTP::
.WORD L\$HW
L\$SPTP::
.WORD L\$SW
L\$LADP::
.WORD L\$LAST
L\$STA::
.WORD 0
L\$CO::
.WORD 0
L\$DTYP::
.WORD 1
L\$APT::
.WORD 0
L\$DTP::
.WORD L\$DISPAT
L\$PRIO::
.WORD PRI07

(5) 002044
(4) 002044 000000
(5) 002046
(4) 002046 000000
(5) 002050
(4) 002050 003
(3) 002051 003
(5) 002052
(4) 002052 000000
(5) 002054 000000
(5) 002056
(4) 002056 000000
(5) 002060
(4) 002060 005512
(5) 002062
(4) 002062 015170
(5) 002064
(4) 002064 000000
(5) 002066
(4) 002066 000000
(5) 002070
(4) 002070 017326
(5) 002072
(4) 002072 017202
(5) 002074
(4) 002074 000000
(5) 002076
(4) 002076 002122
(5) 002100
(4) 002100 104035
(5) 002102
(4) 002102 000000
(5) 002104
(4) 002104 016204
(5) 002106
(4) 002106 017160
(5) 002110
(4) 002110 016776
(5) 002112
(4) 002112 002142
(5) 002114
(4) 002114 000000
(5) 002116
(4) 002116 000000
(5) 002120
(4) 002120 000000
3816
3817 002122
(4) 002122
(3) 002122 052524 034065 050040
(3) 002130 051105 020106 054105
(3) 002136 051105 000
(2) 002142

DESCRIP <TU58 PERF EXER>

L\$ENVI: .WORD 0
L\$EXP1: .WORD 0
L\$MREV: .BYTE C\$REVISI
 .BYTE C\$EDIT
L\$EF: .WORD 0
 .WORD 0
L\$SPC: .WORD 0
L\$DEVP: .WORD L\$DVTYP
L\$REPP: .WORD L\$RPT
L\$EXP4: .WORD 0
L\$EXP5: .WORD 0
L\$AUT: .WORD L\$AU
L\$DUT: .WORD L\$DU
L\$LUN: .WORD 0
L\$DESP: .WORD L\$DESC
L\$LOAD: EMT E\$LOAD
L\$ETP: .WORD 0
L\$ICP: .WORD L\$INIT
L\$CCP: .WORD L\$CLEAN
L\$ACP: .WORD L\$AUTO
L\$PRT: .WORD L\$PRT
L\$TEST: .WORD 0
L\$DLY: .WORD 0
L\$HIME: .WORD 0

L\$DESC: .ASCIZ /TU58 PE

 .EVEN

3819
3820
3821
3822
3823
3824
(3)
3825
3826
3827
3828

002142
002142
000000
002144 177777
002146 177777
002150

```

; **
; THE PROTECT TABLE IS USED BY THE MONITOR TO WARN THE OPERATOR WHEN HE
; TRIES TO TEST THE LOAD DEVICE.
; **
BGNPROT
        .WORD 0           ; DEVICE CSR
        .WORD -1        ; NO MASS BUS
        .WORD -1        ; NO DRIVE
ENDPROT
L$PROT::

```

3835
3836
3837
3838
3839
3840
3841
3842
3843
(4)
(3)
(6)
(6)
(6)
(6)
(6)
(6)
(6)
(6)
(6)
(6)
3844

002150
002150 000011
002152
002152 017330
002154 017532
002156 020004
002160 021376
002162 023002
002164 023772
002166 024556
002170 025546
002172 026332

.SBTTL DISPATCH TABLE

!++
; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
!--

DISPATCH 9

.WORD 9
L\$DISPATCH:;
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9

3852
3853
3854
3855
3856
3857
3858
3859

.SBTTL DEFAULT HARDWARE P-TABLE

; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES.

3860 002174
(3) 002174 000004
(3) 002176
(3) 002176

BGNHW DFPTBL

.WORD L10001-L
L#HW::
DFPTBL::

3861
3862 002176 176500
3863 002200 000300
3864 002202 000003
3865 002204 000000
3866
3872
3873 002206
(3) 002206

.WORD 176500
.WORD 300
.WORD 3
.WORD 0

;CSR ADDRESS
;VECTOR ADDR.
;TEST DRIVE ZERO AND ONE
;NOT PDT TYPE INTERFACE

ENDHW

L10001:

C2

PROGRAM HEADER AND TABLES
CZTUMF.P11 25-JAN-84 08:09

MACY11 30(1046) 25-JAN-84 08:33 PAGE 9-5
SOFTWARE P-TABLE

SEQ 0015

```

3875          .SBTTL  SOFTWARE P-TABLE
3876
3877          ;**
3878          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
3879          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
3880          ;--
3881
3882          BGN$W  SFPTBL
3883
3884          (3) 002206 000010          .WORD  L10002-L
3885          (3) 002210
3886          (3) 002210          SFPTBL::
3887
3888          LENGTH: .WORD  8.          ;TAPE LENGTH
3889          STAEOP: .WORD  1          ;PRINT STATISTICS AT EOP
3890          PR$JF:  .WORD  1          ;PRINT DATA BUF ON COMP. ERROR
3891          CMPDAT: .WORD  1          ;COMPARE DATA
3892          DRVCHK:  .WORD  1          ;ADD DR * TO DATA
3893          EVLTHR:  .WORD  1          ;THRESHOLD FOR EVL TEST
3894          PPSOT9: .WORD  0          ;PRINT UNIT PROTOCOL SUMMARY (TST9)
3895          OT3FL:  .WORD  0          ;TEST 3-DRIVE 0 ONLY FLAG
3896
3897
3898
3899
3900          ENDSW
3901
3902          ENDMQD          L10002:

```

```

3915 .TITLE GLOBAL AREAS
3916 .SBTTL GLOBAL EQUATES SECTION
3944
3954
3955 002230 BGNMOD
3956
3957
3958 ;***
3959 ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
3960 ; ARE USED IN MORE THAN ONE TEST.
3961 ;**
3962 002230 EQUALS
(1) ;
(1) ; BIT DIFINITIONS
(1) ;
(1) 100000 BIT15== 100000
(1) 040000 BIT14== 40000
(1) 020000 BIT13== 20000
(1) 010000 BIT12== 10000
(1) 004000 BIT11== 4000
(1) 002000 BIT10== 2000
(1) 001000 BIT09== 1000
(1) 000400 BIT08== 400
(1) 000200 BIT07== 200
(1) 000100 BIT06== 100
(1) 000040 BIT05== 40
(1) 000020 BIT04== 20
(1) 000010 BIT03== 10
(1) 000004 BIT02== 4
(1) 000002 BIT01== 2
(1) 000001 BIT00== 1
(1) ;
(1) 001000 BIT9== BIT09
(1) 000400 BIT8== BIT08
(1) 000200 BIT7== BIT07
(1) 000100 BIT6== BIT06
(1) 000040 BIT5== BIT05
(1) 000020 BIT4== BIT04
(1) 000010 BIT3== BIT03
(1) 000004 BIT2== BIT02
(1) 000002 BIT1== BIT01
(1) 000001 BIT0== BIT00
(1) ;
(1) ; EVENT FLAG DEFINITIONS
(1) ; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
(1) ;
(1) 000040 EF.START== 32. ; START COMMAND WAS ISSUED
(1) 000037 EF.RESTART== 31. ; RESTART COMMAND WAS ISSUED
(1) 000036 EF.CONTINUE== 30. ; CONTINUE COMMAND WAS ISSUED
(1) 000035 EF.NEW== 29. ; A NEW PASS HAS BEEN STARTED
(1) 000034 EF.PWR== 28. ; A POWER-FAIL/POWER-UP OCCURRED
(1) ;
(1) ;
(1) ; PRIORITY LEVEL DEFINITIONS
(1) ;
(1) 000340 PRI07== 340

```


E2

(1)	000300	PRI06--	300
(1)	000240	PRI05--	240
(1)	000200	PRI04--	200
(1)	000140	PRI03--	140
(1)	000100	PRI02--	100
(1)	000040	PRI01--	40
(1)	000000	PRI00--	0
(1)		;	
(1)		OPERATOR FLAG BITS	
(1)		;	
(1)	000004	EVL--	4
(1)	000010	LOT--	10
(1)	000020	ADR--	20
(1)	000040	IDU--	40
(1)	000100	ISR--	100
(1)	000200	UAM--	200
(1)	000400	BOE--	400
(1)	001000	PNT--	1000
(1)	002000	PRI--	2000
(1)	004000	IXE--	4000
(1)	010000	IBE--	10000
(1)	020000	IER--	20000
(1)	040000	LOE--	40000
(1)	100000	HOE--	100000

3963

```

3976          .SP1TL  ERROR CODE EQUATES
3977
3978          ;THE ERROR CODE OFFSET VALUES :
3979          ;USED BY ROUTINE 'LOG' TO INDEX (BY R5) INTO DEVICE'S DATA BLOCK AND
3980          ;INCREMENT STATISTICS.
3981
3982          000002          SFTRD   **      2
3983          000004          SFTWR   **      4
3984          000006          RCINIT  **      6
3985          000010          OTL     **      8.
3986          000012          OVRN    **     10.
3987          000014          BDCOM   **     12.
3988          000016          HRDRD   **     14.
3989          000020          HRDWR   **     16.
3990          000022          BDCBK   **     18.
3991          000024          SKERR   **     20.
3992          000026          WRLOCK  **     22.
3993          000030          NOMOT   **     24.
3994          000032          CNINIT  **     26.
3995          000034          PARTL   **     28.
3996          000036          NOUNIT  **     30.
3997          000040          CMNDER  **     32.
3998          000042          RECERR  **     34
3999          000044          SLFER   **     36.
4000          000046          SUCOTL  **     38.
4001          000050          TORCVB  **     40.
4002          000054          NCART   **     44.
4003          000056          TOSNDB  **     46.
4004
4005          ;          IN ADDITION, SYSTEM SETUP OR RUNTIME ERRORS ARE:
4006
4007          ;          100.  -          ALL UNITS ABORTED
4008
4009          ;          101.  -          MORE THAN 8. UNITS (16 DRIVES) REQUESTED
4010
4011          ;          102.  -          NEITHER DRIVE SELECTED FOR THIS CONTROLLER
4012
4013          ;          ALL THE ABOVE ARE CLASSIFIED AS SYSTEM FATAL
4014

```

```

4016      ,SBTTL GENERAL EQUATES
4017      ;RADIAL SERIAL CODES:
4018      ;-----
4019      ;THE FLAG BYTE CODES ARE:
4020      000002      RSCMND  == 2           ;"COMMAND" PACKET
4021      000020      RSCONT  == 20          ;"CONTINUE" SINGLE BYTE
4022      000020      RSXON   == 20          ;"XON" SINGLE BYTE
4023      000023      RSXOFF  == 23          ;"XOFF" SINGLE BYTE
4024      000004      RSINIT  == 4           ;"INIT" SINGLE BYTE
4025      000001      RSDATA  == 1           ;"DATA" PACKET
4026      000002      RSEND   == RSCMND     ;"END" PACKET FLAG IS "COMMAND"
4027      ;-----
4028      ;END PACK SIZE:
4029      000016      RSNDSZ  == 14.         ;TOTAL BYTES IN COMMAND PACKET
4030      ;MESSAGE PACK SIZE:
4031      000012      RSMSIZ  == 12          ;10. BYTES FOR BYTE COUNT INSIDE CMND PACK
4032      ;DATA PACK SIZE:
4033      000204      RSDASZ  == 132.        ;TOTAL BYTES IN DATA PACKET
4034      ;DATA * END PACK SIZE:
4035      000222      RSDNSZ  == RSDASZ+RSNDSZ
4036      ;GET CHARACTERISTICS DATA PACKET SIZE
4037      000034      RSGCDP  == 28.         ;TOTAL BYTES FOR GET CHAR DATA PACKET
4038      ;MINUS THE END PACKET
4039      000016      RSSNSZ  == RSMSIZ * 4   ;SIZE FOR SENDING COMMAND PACK
4040      001036      RCBFSZ  == 4*RSDASZ+RSNDSZ ;4 DATA PAKS AND END PACK
4041      ;IS SIZE OF RCV BUFFERS
4042      ;-----
4043      ;THE OP CODES ARE:
4044      000100      RSSEND  == 100          ;END PACK DESCRIPTOR
4045      000003      RSSWR   == 3           ;WRITE
4046      000002      RSSRD   == 2           ;READ
4047      000005      RSSSEK  == 5           ;SEEK
4048      000012      RSSGET  == 12          ;GET CHARACTERISTICS
4049      000000      RSSNOP  == 0           ;NO-OPERATION
4050      000001      RSSNIT  == 1           ;INITIALIZE
4051      000007      RSSSLF  == 7           ;SELF TEST
4052      ;-----
4053      ;THE SUCCESS CODES ARE:
4054      177720      ESABO   ==-48.         ;BAD COMMAND FROM HOST
4055      177767      ESNCRT  ==-9.          ;NO CARTRIDGE
4056      177770      ESNONX  ==-8.          ;NO DRIVE
4057      000000      ESOK    ==0            ;OP COMPLETE SUCCESS
4058      177776      ESPART  ==-2          ;PARTIAL OP
4059      177740      ESSK    ==-32.         ;SEEK ERROR
4060      000001      ESTRY   ==1           ;RETRY OCCURRED
4061      177765      ESWLOC  ==-11.         ;WRITE PROTECTED
4062      177737      ESNOMO  ==-33.         ;MOTOR STOPPED
4063      177720      ESCMD   ==-48.         ;COMMAND ERROR
4064      177711      ESREC   ==-53.         ;BAD RECORD NUMBER.
4065      177757      ESCKS   ==-17.         ;TU CHKSUM ERROR
4066      177777      ESSLF   ==-1.          ;SELF TEST ERROR
4067      177757      ESCKSM  =ESCKS
4068      177757      ESWR    =ESCKS
4069      177757      ESRD    =ESCKS
4070      ;-----

```

4072			.SBTTL ERROR MESSAGE DESCRIPTIONS
4073			
4074			;THE TABLE OF ERROR MESSAGES (ADDRESSES). ABNDX(R5) CONTAINS THE OFFSET
4075			;OF THE REASON. IT'S ABSOLUTE ADDRESS IS RSNTAB + ABNDX(R5).
4076			
4077	002230	002324	RSNTAB: MSNLOG
4078	002232	003056	MSSFRD
4079	002234	003116	MSSFWR
4080	002236	002540	MSRNIT
4081	002240	003002	MSQRSP
4082	002242	003262	MSOVRN
4083	002244	002406	MSCOM
4084	002246	003156	MSHORD
4085	002250	003220	MSHDWR
4086	002252	002560	MSHCHK
4087	002254	002310	MSSKER
4088	002256	002516	MSWPRO
4089	002260	002450	MSNOMO
4090	002262	002622	MSNIT
4091	002264	002636	MSPART
4092	002266	002660	MSUNIT
4093	002270	002706	MSCMD
4094	002272	002722	MSREC
4095	002274	002366	MSELF
4096	002276	002742	MSWRSP
4097	002300	002766	MSNRSP
4098	002302	002324	MSNLOG
4099	002304	002466	MSNOTP
4100	002306	003034	MSTOSN

```

4102                                     ;HERE ARE THE MESSAGES PROPER;
4103
4104 002310 042523 045505 042440  MSSKER:: .ASCIZ /SEEK ERROR/                ;DEVICE COULD NOT READ HEADER
4105                                     .EVEN
4106 002324 054523 052123 046505  MSNLOG:: .ASCIZ /SYSTEM ERROR/                ;DIAGNOSTIC HUNG. BETTER RE-BOOT
4107                                     .EVEN
4108 002342 040502 020104 040504  MSBDA:: .ASCIZ /BAD DATA IN PACKET/          ;HOST DATA CHECK FOUND ERROR, DEVICE MAY
4109                                     .EVEN                ;HAVE READ CORRECTLY.
4110 002366 042523 043114 052040  MSSELF:: .ASCIZ /SELF TEST ERROR/            ;MICRO DIAGNOSTIC FAILED, BUT DEVICE COU
4111                                     .EVEN                ;SEND AN END PACKET.
4112 002406 040502 020104 040504  MSCOM:: .ASCIZ /BAD DATA W-O DATA CHECK ERR AT TU/ ;PREVIOUS DATA CHECK
4113                                     .EVEN                ;ERROR NOT DUE TO DEVICE READ OPERATION
4114 002450 047515 047524 020122  MSNOMO:: .ASCIZ /MOTOR STOPPED/                ;DEVICE COULD NOT GET ANY MEANINGFUL SIG
4115                                     .EVEN                ;FROM TAPE.
4116 002466 040503 052122 044522  MSNOTP:: .ASCIZ /CARTRIDGE NOT IN PLACE/        ;NO MEDIA OR BAD SWITCH
4117                                     .EVEN
4118 002516 051127 052111 020105  MSWPRO:: .ASCIZ /WRITE PROTECTION/            ;CARTRIDGE WRITE PROTECT TAB MISSING OR
4119                                     .EVEN                ;SWITCH BAD
4120 002540 042522 044503 053105  MSRNIT:: .ASCIZ /RECIEVING INIT/              ;DEVICE SENT INIT REQUEST
4121                                     .EVEN
4122 002560 047510 052123 043040  MSHCHK:: .ASCIZ /HOST FOUND PACKET CHECKSUM ERROR/ ;DEVICE SENT PACK WITH
4123                                     .EVEN                ;BAD CHECKSUM
4124 002622 040503 023516 020124  MSNIT:: .ASCIZ /CAN'T INIT/                    ;DEVICE SENT BYTE OTHER THAN "CONTINUE"
4125                                     .EVEN                ;DURING INITIALIZATION
4126 002636 040520 052122 040511  MSPART:: .ASCIZ /PARTIAL OPERATION/            ;END OF MEDIUM ENCOUNTERED
4127                                     .EVEN
4128 002660 047042 047117 042455  MSUNIT:: .ASCIZ /"NON-EXISTENT" DRIVE/        ;DEVICE RECV'D TOO LARGE DRIVE NUMBER
4129                                     .EVEN
4130 002706 040502 020104 047503  MSCMD:: .ASCIZ /BAD COMMAND/                  ;DEVICE COULD NOT UNDERSTAND HOST
4131                                     .EVEN
4132 002722 040502 020104 042522  MSREC:: .ASCIZ /BAD RECORD NO./              ;DEVICE RECV'D TOO LARGE A RECCRD NUMBER
4133                                     .EVEN
4134 002742 051127 047117 020107  MSWRSP:: .ASCIZ /WRONG SUCCESS CODE/          ;HOST COULD NOT DECIPHER CODE IN END PAC
4135                                     .EVEN
4136 002766 047516 051040 051505  MSNRSP:: .ASCIZ /NO RESPONSE/                ;TIME OUT WAITING FOR BYTE IN RCV BUF ON
4137                                     .EVEN
4138 003002 047111 042504 044503  MSQRS?: .ASCIZ \INDECIPHERABLE FLAG BYTE\ ;HOST COULD NOT UNDERSTAND 1ST BYTE
4139                                     .EVEN                ;RESPONSE FROM TU AS PROPER PROTOCOL
4140 003034 044524 042515 047440  MSTOSN:: .ASCIZ /TIME OUT ON SEND/            ;DLV 'READY' NEVER WENT HIGH
4141                                     .EVEN
4142 003056 042522 047503 027126  MSSFRD:: .ASCIZ /RECOV. DATA CHECK ERR ON RD OP/ ;TU58 RESPONDED WITH "DATA-CHE
4143                                     .EVEN                ;ERROR ON READ OP. ;HOST RETRY(S) SUCCE
4144 003116 042522 047503 027126  MSSFWR:: .ASCIZ /RECOV. DATA CHECK ERR ON WR OP/ ;SAME BUT WR OR WR VERIFY OPER
4145                                     .EVEN
4146 003156 047125 042522 047503  MSHDRD:: .ASCIZ /UNRECOV. DATA CHECK ERR ON RD OP/ ;TU58 RESPONDED WITH "DATA-C
4147                                     .EVEN                ;ERROR ON READ OP. ;RETRIES UNSUCCESSFU
4148 003220 047125 042522 047503  MSHDWR:: .ASCIZ /UNRECOV. DATA CHECK ERR ON WR OP/ ;SAME BUT WR OPERATION
4149                                     .EVEN
4150 003262 046104 020126 051105  MSOVRN:: .ASCIZ /DLV ERROR IN RECEIVE/        ;DLV ERROR (THE CONTENTS PRINTED OUT)
4151                                     .EVEN

```


4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241

000000
000002
000004

000020
000022
000024
000026
000030
000032
000034
000036

000060
000062
000064

000066
000070
000072
000074
000076
000100

000102
000104
000106
000110
000112
000114
000116

.S3TTL DATA BLOCK FORMAT

;R5 --> TOP OF 1 OF THE 8 DATA BLOCKS (1 PER UNIT) DURING EXECUTION
;R5 IS THE STATUS WORD CONTAINING:

STATUS	..	0.
RETRY	..	2.
ABNDX	..	4.
;R0		
;R1		
;R2		
;R3		
;R4		
TSTPC	..	16.
RCSR	..	18.
RCDB	..	20.
XMSR	..	22.
XMDB	..	24.
XSPKMM	..	26.
XSFLG	..	28.
XSCNT	..	30.
	BLKW	8.
DR	..	48.
TRK	..	50.
REC	..	52.
TMP	..	54.
SNDCNT	..	56.
PATTEN	..	58.
DLV	..	60.
SUCCS	..	62.
CMDSNT	..	64.
RCVBUF	..	66.
PKPTR	..	68.
XSPTX	..	70.
WRTNO	..	72.
WRTN1	..	74.
RDNO	..	76.
RDN1	..	78.

;BIT15 = ABORTED
;BIT14 = SEND "BREAK"
;BIT13 = RETRY FLAG BYTE ERROR (DATA PACKS)
;BIT12 = TEMP STOR WRITE MACRO
;BIT11 = UNIT NOT BEING TESTED
;BIT10 = RETRYING DATA ERROR
;BIT9 = TUS8 CHKSUM ERROR
;BIT8 = RD/WR OPERATION
;BIT7 = NORMAL/REDUCED THRESHOLD (MACROS)
;BIT6 = HOST DATA COMPARE ERROR
;BIT5 = WR VERIFY OPERATION
;BIT4 = TYPE OF PAK SENT (DATA 1CMD)
;BIT3 = RETRY FLAG BYTE ERR.(SEND COMMAND PACK)
;BIT0,1,2=UNIT NO.
;DEVICE STATE
;# OF RETRIES
;ERROR NUMBER FOR LOG
;STORAGE FOR REGISTERS USED IN TEST BODY
;STORED WITH SWAPOW
;RETRIEVED WITH SWAPIN
;
;
; POINTER TO NEXT EXECUTABLE TEST INST.
;DLV RCV STATUS ADDRESS
;DLV RCV DATA ADDRESS
;DLV SND STATUS ADDRESS
;DLV SND DATA ADDRESS
;THE NUMBER OF PACKETS TO RECEIVE
;THE EXPECTED FLAG OF 1ST PACKET
;THE EXPECTED COUNT OF 1ST PACKET
;FOR MULTIPLE PACKET RECIEVES (MAX.4)
;CONSECUTIVE XSFLGS AND XSCNTS
;DR==0 OR 1; BIT8,9 DRIVE SELECTED BY OPERATOR
;COUNTER FOR TRACK NUMBER
;RECORD (BLOCK #)
;
;TEST MACRO REGISTER
;THE # OF BYTES FOR SENDING PACKET
;DATA PATTERN-LOWER BYTE USED
;CONTENTS OF RCDB ON DLV ERROR
;SUCCESS CODE OF LAST END PACKET
;TYPE OF COMMAND CURRENT IN EVEN BYTE; 9BIT15=-VE
;
; POINTER TO 542. BYTE BUFFER (4 DATA PAKS + END
; POINTER TO TOP OF PACKET
; POINTER TO CURRENTLY USED XSFLG OR XSCNT
; THE # OF 512. BYTE BLOCKS WRITTEN DRO
; THE # OF 512. BYTE BLOCKS WRITTEN DR1
; THE # OF 512. BYTE BLOCKS READ DRO
; THE # OF 512. BYTE BLOCKS READ DR1

```

4243 ;AND THE ERROR LOG...
4244 ;SPLIT INTO A BYTE PER DRIVE: ; DR1 ! DR0 !
4245 ;
4246 ;
4247 ;
4248 ;-----;
4249 ;OFFSET IN DATA BLOCK ;ERROR TYPE ;ERRCODE;MSG CODE;SUC. CODE
4250 ;-----;
4251 000120 LGOFFST == 80. ;**RESERVED**
4252 000122 SOFTR == 82. ;SOFT READ ;SFTRD ;MSSFRO ;ESCKSM
4253 000124 SOFTW == 84. ;SOFT WRITE ;SFTWR ;MSSFWR ;ESSKSM
4254 ; WORD ;RECEIVED INIT ;RCINIT ;MSKNIT ;*****
4255 ; WORD ;BAD FLAG BYTE ;OTL ;MSQRSP ;*****
4256 ;
4257 ; THEN THOSE CODES WHICH HAVE N TRIES BEFORE ABORT
4258 ;
4259 000132 T4TRY == 90. ;DLV ERROR ;OVRN ;MSOVRN ;*****
4260 000134 BDATA == 92. ;BAD DATA ;BDCOM ;MSDATA ;*****
4261 000136 HARDR == 94. ;HARD READ ;HRDRD ;MSHRDR ;ESCKSM
4262 000140 HARDW == 96. ;HARD WRITE ;HRDWR ;MSHDWR ;ESCKSM
4263 ; WORD ;CHKSM AT HOST ;BDCHK ;MSHCHK ;*****
4264 ; WORD ;SEEK ERROR TOTAL ;SKERR ;MSSKER ;*****
4265 000146 T1TRY == 102. ;WRITE PROTECT ;WRLOCK ;MSWPRO ;ESWLCK
4266 ; WORD ;NO MOTOR ;NOMOT ;MSNOMO ;ESNOMO
4267 ; WORD ;CANT INIT ;CNINIT ;MSNIT ;*****
4268 ; WORD ;PARTIAL OP ;PARTL ;MSPART ;ESPART
4269 ; WORD ;NO UNIT ;NOUNIT ;MSUNIT ;ESNONX
4270 ; WORD ;COMMAND ERROR ;CMNDR ;MSCMD ;ESCMD
4271 ; WORD ;BAD RECORD NO ;RECERR ;MSREC ;ESREC
4272 ; WORD ;SELF TEST ERROR ;SLFER ;MSSELF ;*****
4273 ; WORD ;WRONG SUC.CODE ;SUCOTL ;MSWRSP ;*****
4274 ; WORD ;NO RESPONSE ;TORCVB ;MSNRSP ;*****
4275 ; WORD ;**RESERVED**
4276 ; WORD ;NO CARTRIDGE ;NOCART ;MSNOTP ;ESNCRT
4277 ; WORD ;TIME OUT SEND ;TOSNDB ;MSTOSN ;*****
4278 ;
4279 ;
4280 000202 BLKEND == 130. ;OFFSET OF END OF STATISTICS (RESERVED)
4281 000204 TUVECT == 132. ;VECTOR ADDRESS
4282 000206 SAVCNT == 134. ;BYTE COUNT SAVED DURING RETRY ON WRITE OPERATIO
4283 000210 MRSP == 136. ;***** FLAG INDICATING MRSP
4284 000212 BLKSIZ == 138. ;** RESERVED **
4285 ;

```



```

4288 .SBTTL DEVICE DATA BLOCK ALLOCATION
4289
4290
4291 ;TABLE OF DEVICE DATA BLOCK ADDRESSES
4292
4293
4294 003352 003372 BLKTBL:: .WORD DEVO
4295 003354 003604 .WORD DEV1
4296 003356 004016 .WORD DEV2
4297 003360 004230 .WORD DEV3
4298 003362 004442 .WORD DEV4
4299 003364 004654 .WORD DEV5
4300 003366 005066 .WORD DEV6
4301 003370 005300 LSTDEV:: .WORD DEV7
4302
4303
4304 ;AND STORAGE FOR EACH:
4305
4306 003372 000212 DEVO: .BLKB BLKSIZ
4307 003604 000212 DEV1: .BLKB BLKSIZ
4308 004016 000212 DEV2: .BLKB BLKSIZ
4309 004230 000212 DEV3: .BLKB BLKSIZ
4310 004442 000212 DEV4: .BLKB BLKSIZ
4311 004654 000212 DEV5: .BLKB BLKSIZ
4312 005066 000212 DEV6: .BLKB BLKSIZ
4313 005300 000212 DEV7: .BLKB BLKSIZ

```

4329
4330
4331
4332
4333
4334
(4)
(3)
(3)
(3)
(2)
4335
4347
4348
4366

.SBTTL GLOBAL TEXT SECTION

; NAMES OF DEVICES SUPPORTED BY PROGRAM

DEV TYP <TU58 CONTROLLER>

005512
005512
005512 052524 034065 041440
005520 047117 051124 046117
005526 042514 000122

L\$DVTYP: .ASCIZ /TU58 CO

.EVEN

```

4375          .SBTTL  SYSTEM MACRO DEFINITIONS
4376
4377          .MACRO  PUSH  ,REG
4378
4379          .NLIST
4380          .LIST ME
4381          .LIST
4382
4383
4384
4385
4386          .NLIST
4387          .NLIST ME
4388          .LIST
4389          .ENDM
4390
4391          .MACRO  POP,REG
4392
4393          .NLIST
4394          .LIST ME
4395          .LIST
4396
4397
4398
4399
4400
4401
4402
4403          ;**
4404          ;THE MACRO 'SWAPIN' RETRIEVES THE TEST REGISTERS WHICH WERE SAVED
4405          ;IN THE DEVICE DATA BLOCK.
4406          ;**
4407
4408          .MACRO  SWAPIN
4409
4410          .NLIST
4411          .LIST ME
4412          .LIST
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424          ;**
4425          ;THE MACRO 'SWAPOW' SAVES THE CURRENT STATE OF THE UNIT IN THE DRIVE
4426          ;DATA BLOCK IN SO THAT THE SCHEDULER MAY 'SWAPIN' ANOTHER UNIT.
4427          ;**
4428
4429          .MACRO  SWAPOW
4430          .NLIST

```

```

MOV      REG, -(SP)

```

```

MOV      (SP)+,REG

```

```

MOV      6.(R5),R0
MOV      8.(R5),R1
MOV      10.(R5),R2
MOV      12.(R5),R3
MOV      14.(R5),R4

```

C3

GLOBAL AREAS MACY11 30(1046) 25-JAN-84 08:33 PAGE 12-1
CZTUUF.P11 25-JAN-84 08:09 SYSTEM MACRO DEFINITIONS

SEQ 0028

4431	.LIST ME		
4432	.LIST		
4433		MOV	R0,6.(R5)
4434		MOV	R1,8.(R5)
4435		MOV	R2,10.(R5)
4436		MOV	R3,12.(R5)
4437		MOV	R4,14.(R5)
4438			
4439	.NLIST		
4440	.NLIST ME		
4441	.LIST		
4442	.ENDM		

4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500

```
***  
;THE WRITE MACRO IMPLEMENTS THE COMPLETE PROTOCOL NECESSARY TO BUILD  
;A COMMAND PACKET AND SUBSEQUENT DATA PACKETS (UNTIL THE BYTE COUNT  
;(BCNT) IS SATISFIED).  
;  
;SETS UP THE EXPECTED PROTOCOL RESPONSES: THE NUMBER OF PACKETS  
;(XSPKMN) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT). CALLS  
;'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET  
;CHECKSUM.  
;  
; INPUTS - DEVICE BLOCK DR5  
;          TRBUF - BUFFER ADDRESS  
;          UNIT'S TEST REGISTERS FROM 'SWAPIN'  
; OUTPUTS - SNOCNT(R5) = # OF BYTES TO SEND  
;          XSPKMN = # OF PACKETS EXPECTED  
;          XSFLG = FLAG BYTE OF 1ST PACKET  
;          XSCNT = BYTE COUNT OF 1ST PACKET  
;          . ***  
;          * SUBSEQUENT XSFLGS  
;          . >  
;          * AND XSCNTS  
;          . ***  
;--
```

.MACRO TUWRIT PTRN,REC,BCNT,DR,VER,?A,?B,?C,?D,?E,?F,?G,?H,?T

.NLIST
.LIST ME
.LIST

```
T:      MOV      @TRBUF,R0      ;MAKE COMMAND PACKET:  
        MOVB     @RSCMND,@R0   ;COMMAND FLAG  
        MOVB     @RSMSIZ,1(R0) ;THIS SIZE  
        MOVB     @RSSWR,2(R0)  ;INSERT OP CODE-WRITE  
        MOVB     VER,3.(R0)    ;VERIFY (1 OR 0)  
        MOVB     DR,4.(R0)     ;DRIVE #  
        MOVB     @O20,5.(R0)   ;MAINTENANCE MODE SWITCH  
        CLR      6.(R0)        ;NO SEQUENCE #  
        MOV      BCNT,8.(R0)   ;TOTAL COUNT TO WRITE  
        MOV      REC,10.(R0)   ;AT RECORD N  
        MOV      @RSMSIZ,R1    ;THE PACKET SIZE PLUS 2  
        TST     (R1)          ;(FLAG AND COUNT) INTO R  
        MOV      @RSSNSZ,SNOCNT(R5) ;LOAD THE SIZE TO S  
        CALL    CHKSUM        ;RO --> R1-COUNT  
        MOV      R1,(R0)      ;PUT CHKSUM IN PACKET  
;SET UP EXPECTATIONS:  
        MOV      @RSCONT,XSFLG(R5) ;THE FLAG  
        MOV      @1,XSCNT(R5)   ;THE COUNT  
        MOV      @1,XSPKMN(R5) ;THE # PACKETS EXPECTED  
        MOV      BCNT,R2       ;GET # OF DATA BYTES  
        CALL    RSVP          ;SEND (AND RETURN TO SCH  
        BIT     @BIT3,@R5      ;FLAG BYTE ERROR?  
        BNE     T              ;YES  
        BIC     @BIT12,@R5     ;FLAG FOR LAST PACKET  
A:      MOV      @TRBUF,R0     ;POINT TO TOP OF BUFFER  
        CMP     R2,@128.      ;START DATA PACKET(S)  
        BHI     B              ;BCNT > 128.!
```

4501
 4502
 4503
 4504
 4505
 4506
 4507
 4508
 4509
 4510
 4511
 4512
 4513
 4514
 4515
 4516
 4517
 4518
 4519
 4520
 4521
 4522
 4523
 4524
 4525
 4526
 4527
 4528
 4529
 4530
 4531
 4532
 4533
 4534
 4535
 4536
 4537
 4538
 4539
 4540
 4541
 4542
 4543
 4544
 4545
 4546
 4547
 4548
 4549

```

MOV R2,R1 ;BCNT<128.
BIS #BIT12,R5 ;SO LAST PACKET NOW
BR C ;USE REMAINING COUNT
B: MOV #128.,R1 ;USE 128. BYTES
C: MOVB R1,1(R0) ;COPY COUNT TO BUFFER
MOV R1,R3 ;R3=COUNTER TO LOAD BUFF
MOVB #RSDATA,R0 ;FLAG FIRST
TST (R0)+ ;SKIP COUNT
D: MOVB PTRN,(R0)+ ;INSERT DATA
DEC R3 ;MORE?
BIII D ;YES
MOV #TRBUF,R0 ;-->TOP AGAIN
MOVB 1(R0),R1 ;GET COUNT
BIC #177400,R1 ;ZERO SIGN EXTEND
MOV R1,SNDcnt(R5) ;HOW MANY TO SEND PLUS
ADD #4,SNDcnt(R5) ;FLAG,COUNT,CHKSUM
ADD #2,R1 ;COMPENSATE FOR FLAG + C
CALL CHKSUM ;FOR CHKSUM CALC.
MOVB R1,(R0)+ ;CHKSUM INTO PACKET
SWAB R1 ;EVEN ON AN ODD
MOVB R1,(R0)+ ;BYTE BOUNDARY
BIT #BIT12,R5 ;LAST DATA PACKET?
BEQ E ;NO
MOV #RSEND,XSFLG(R5) ;YES-EXPECT 'END'
MOV #RSNDsz,XSCNT(R5) ;OF THIS SIZE
MOV #1,XSPKnm(R5) ;AND 1 PACKET
BR F ;SEND
E: MOV #RSCONT,XSFLG(R5) ;(NOT LAST), EXPECT '
MOV #1,XSCNT(R5) ;AND 1 BYTE
F: MOV #1,XSPKnm(R5) ;AND 1 PACKET
CALL RSVP ;SEND PACKET
;AND RETURN TO SCHEDULER
BIT #BIT3,R5 ;FLAG BYTE RETRY?
BNE T ;YES
BIT #BIT10,R5 ;RETRY DATA ERROR?
BNE G ;YES
SUB #128.,R2 ;NO, MORE DATA TO SEND?
BHI A ;YES
BR H ;NO
G: TURTRY REC,BCNT,DR ;RETRY HERE
BIT #BIT10!BIT3,R5 ;RETRY AGAIN?
BNE G ;YES
H: NOP ;DONE
  
```

.NLIST
 .NLIST ME
 .LIST
 .ENDM

```

4552      ;
4553      ; THE SEEK MACRO IMPLIMENTS THE COMPLETE PROTOCOL TO INITIATE A SEEK
4554      ; SEQUENCE.
4555      ;
4556      ; SETS UP THE EXPECTED PROTOCOL RESPONSES: THE NUMBER OF PACKETS
4557      ; (XSPKIM) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT). CALLS
4558      ; 'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
4559      ; CHECKSUM.
4560      ;
4561      ; INPUTS - DEVICE BLOCK BR5
4562      ;           UNITS TEST REGISTERS FROM SWAPIN
4563      ;           TRBUF - BUFFER ADDRESS
4564      ;
4565      ; OUTPUTS -
4566      ;           XSPKIM = # OF PACKETS EXPECTED
4567      ;           XSFLG = FLAG BYTE OF 1ST PACKET
4568      ;           XSCNT = BYTE COUNT OF 1ST PACKET
4569      ;           . ***
4570      ;           . * SUBSEQUENT XSFLGS
4571      ;           . *
4572      ;           . * AND XSCNTS
4573      ;           . ***
4574      ;
4575      ;
4576      ;
4577      ;
4578      ;
4579      ;
4580      ;
4581      ;
4582      ;
4583      ;
4584      ;
4585      ;
4586      ;
4587      ;
4588      ;
4589      ;
4590      ;
4591      ;
4592      ;
4593      ;
4594      ;
4595      ;
4596      ;
4597      ;
4598      ;
4599      ;
4600      ;
4601      ;
4602      ;
4603      ;
4604      ;
4605      ;
4606      ;
4607      ;

```

```

.MACRO TUSEEK REC,DR,?A

.NLIST
.LIST ME
.LIST

      A:      MOV      #TRBUF,RO      ;-->(POINT TO) XMIT BUFF
              MOV      #RSCMND,RO    ;FORM COMMAND MESSAGE PA
              MOV      #RSMSIZ,1(RO) ;THIS BIG
              MOV      #RESSEK,2(RO) ;OP CODE IS SEEK
              MOV      REC,10.(RO)   ;TO THIS RECORD
              MOV      DR,4.(RO)     ;AND WHICH DRIVE
              CLRB     3.(RO)        ;NO MODIFIER
              CLRB     5.(RO)        ;NO SWITCHES
              CLR      6.(RO)        ;NO SEQUENCE #
              CLR      8.(RO)        ;NO BYTE COUNT
              MOV      #RSMSIZ,R1     ;GET COUNT
              TST      (R1).         ;PLUS FLAG * BCNT
              ;FOR CHECKSUM CALC
              CALL     CHKSUM        ;RO-->TOP R1=# OF BYTE
              MOV      R1,(RO)       ;INSERT INTO PACKET
              ;SET UP EXPECTATIONS:
              MOV      #RSSNSZ,SNDcnt(R5) ;HOW MANY TO SEND
              MOV      #RSCMND,XSFLG(R5) ;EXPECT END PACK
              MOV      #RSNDSZ,XSCNT(R5) ;COUNT WITH THIS
              MOV      #1,XSPKIM(R5) ;EXPECT ONLY 1 PACKET

              CALL     RSVP          ;SEND
              ;AND RETURN TO SCHEDULER
              BIT      #BIT3,BR5     ;RETRY (FLAG BYTE ERROR)
              BNE     A              ;YES

```

G3

GLOBAL AREAS MACY11 30(1046) 25-JAN-84 08:33 PAGE 14-1
CZTUUF.P11 25-JAN-84 08:09 SYSTEM MACRO DEFINITIONS

SEQ 0032

4608
4609
4610
4611
4612

.NLIST
.NLIST ME
.LIST
.ENDM


```

4615      ;**
4616      ;THE RETRY MACRO IMPLMENTS THE COMPLETE PROTOCOL NECESSARY TO INITIATE
4617      ;A RETRY (READ OPERATION) SEQUENCE.
4618      ;
4619      ;SETS UP THE EXPECTED PROTOCOL RESPONSES: THE NUMBER OF PACKETS
4620      ;(XSPKNM) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT). CALLS
4621      ;'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
4622      ;CHECKSUM.
4623      ;
4624      ; INPUTS - DEVICE BLOCK ORS
4625      ;          TRBUF - BUFFER ADDRESS
4626      ;          UNITS TEST REGISTERS FROM SWAPIN
4627      ;
4628      ; OUTPUTS - SNDCNT(R5) = # OF BYTES TO SEND
4629      ;          XSPKNM = # OF PACKETS EXPECTED
4630      ;          XSFLG = FLAG BYTE OF 1ST PACKET
4631      ;          XSCNT = BYTE COUNT OF 1ST PACKET
4632      ;          . ***
4633      ;          . * SUBSEQUENT XSFLGS
4634      ;          . >
4635      ;          . * AND XSCNTS
4636      ;          . ***
4637      ;--
4638
4639      .MACRO TURTRY REC,BCNT,DR,?A,?B,?C,?D,?E
4640
4641      .NLIST
4642      .LIST ME
4643      .LIST
4644
4645      D:      MOV      #TRBUF,R0      ;FORM CMD PACK:
4646      MOVB     #RSCMD,OR0      ;MESSAGE PACK TYPE
4647      MOVB     #RMSIZ,1(R0)     ;THIS BIG
4648      MOVB     #RSSRD,2(R0)     ;OP CODE-READ
4649      MOV      REC,10.(R0)      ;THIS RECORD
4650      MOVB     DR,4.(R0)        ;THIS DRIVE
4651      CLRB     3(R0)            ;PRESET NORM THRESHOLD
4652      TSTB     OR5              ;REDUCED?
4653      BPL      E                ;NO
4654      INCB     3(R0)            ;YES-CHANGE THRESHOLD
4655      E:      MOV      BCNT,8.(R0) ;# BYTES DESIRED
4656      MOVB     #O20,5.(R0)     ;MAINTENANCE MODE
4657      CLR      6.(R0)           ;NO SEQUENCE #
4658      MOV      #RMSIZ,R1        ;SIZE OF PACKET
4659      TST      (R1)+            ;PLUS FLAG+COUNT INTO R1
4660      MOV      #RSSNS?,SNDCNT(R5) ;SET UP SIZE TO SEND
4661
4662      CAL     CHKSUM            ;FORM CHECKSUM R1-COUNT
4663      MOV      R1,(R0)          ;INSERT IN PACKET
4664
4665      MOV      BCNT,R1          ;SET EXPECTATIONS:
4666      MOV      #XSFLG,R3        ;CALC # OF DATA PACKETS
4667      ADD      R5,R3            ;OFFSET OF FLAG
4668      CLR      R2                ;ABS. ADDR. OF XSFLG
4669      A:      INC      R2          ;PRESET
4670      ;# PACKETS EXPECTED

```

4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687

	MOV	#RSDATA,(R3)+	;LOAD XSFLG
	MOV	#132.,(R3)+	;AND EXPECT COUNT
	SUB	#128.,R1	;NEG RESULT LAST TIME
	BLOS	C	;LAST TIME!
	BR	A	;MORE TO DO
C:	INC	R2	;ADD ONE FOR END PACK
	MOV	R2,XSPKMM(R5)	;SAVE # PACKETS TO EXPECT
	MOV	#RSEND,(R3)+	;EXPECT AN END
	MOV	#RSNDSZ,(R3)	;THIS BIG-14. BYTES
	CALL	RSVP	;SEND
			;AND RETURN TO SCHEDULER

.NLIST
.NLIST ME
.LIST
.ENDM

4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745

```

;***
;THE READ MACRO IMPLMENTS THE COMPLETE PROTOCOL NECESSARY TO INITIATE
;A READ SEQUENCE.
;
;SETS UP THE EXPECTED PROTOCOL RESPONSES: THE NUMBER OF PACKETS
;(XSPKNM) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT). CALLS
;'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
;CHECKSUM.
;
; INPUTS - DEVICE BLOCK DR5
;          TRBUF - BUFFER ADDRESS
;          UNITS TEST REGISTERS FROM SWAPIN
;
; OUTPUTS - SNDCNT(R5) = # OF BYTES TO SEND
;           XSPKNM = # OF PACKETS EXPECTED
;           XSFLG = FLAG BYTE OF 1ST PACKET
;           XSCNT = BYTE COUNT OF 1ST PACKET
;
;           . ***
;           . * SUBSEQUENT XSFLGS
;           . >
;           . * AND XSCNTS
;           . ***
;--

```

.MACRO TUREAD REC,BCNT,DR,VER,?A,?B,?C,?D,?E

.NLIST
.LIST ME
.LIST

```

E:      MOV     #TRBUF,R0      ;FORM CMND PACK:
        MOVB   #RSCMND,R0    ;MESSAGE PACK TYPE
        MOVB   #RSMSIZ,1(R0) ;THIS BIG
        MOVB   #RSSRD,2(R0)  ;OP CODE IS READ
        MOV    REC,10(R0)    ;THIS RECORD
        MOVB   DR,4(R0)      ;THIS DRIVE
        MOVB   VER,3(R0)     ;VERIFY
        MOV    BCNT,8(R0)    ;TOTAL BYTES TO READ
        MOVB   #020,5(R0)    ;MAINTENANCE MODE
        CLR    6(R0)         ;NO SEQUENCE #
        MOV    #RSMSIZ,R1    ;GET SIZE OF PACKET
        TST    (R1)+         ;+2 FOR CHECKSUM
        MOV    #RSSNSZ,SNDCNT(R5) ;SIZE TO SEND
        CALL   CHKSUM        ;FORM CHECKSUM R1=COUNT
        MOV    R1,(R0)       ;INSERT CHECKSUM

        MOV    BCNT,R1      ;SET EXPECTATIONS:
                                ;CALC # OF DATA PACKETS
                                ;GET OFFSET
                                ;ABS. ADDR. OF XSFLG
                                ;PRESET AS NONE
                                ;# PACKETS EXPECTED
A:      INC    R2
        MOV    #RSDATA,(R3)+ ;LOAD XSFLG
        MOV    #132,(R3)+   ;AND EXPECTED COUNT
        SUB    #128,,R1     ;NEG RESULT LAST TIME

```

4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764

	BLOS	C	;LAST TIME
	BR	A	;MORE TO DO
C:	INC	R2	;ADD ONE FOR END PACK
	MOV	R2,XSPKMM(R5)	;SAVE # PACKETS TO EXPECT
	MOV	#RSEND,(R3)	;EXPECT AN END ALSO...
	MOV	#RSNDSZ,(R3)	;THIS BIG-14. BYTES
	CALL	RSVP	;SEND
			;AND RETURN TO SCHEDULER
D:	BIT	#BIT10!BIT3,R5	;RETRY?
	BEQ	B	;NO.
	TURTRY	REC,BCNT,DR	;YES
	BR	D	;ANOTHER RETRY?
B:	NOP		;NO

.NLIST
.NLIST ME
.LIST
.ENDM

4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822

```

; **
; THE SELF TEST MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO
; INITIATE A 'DIAGNOSE' SEQUENCE.
;
; SETS UP THE EXPECTED PROTOCOL RESPONSES; THE NUMBER OF PACKETS
; (XSPKMN) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT). CALLS
; 'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
; CHECKSUM.
;
; INPUTS - DEVICE BLOCK &R5
;          TRBUF - BUFFER ADDRESS
;          UNITS REGISTERS TEST FROM SWAPIN
;
; OUTPUTS - SNDCNT(R5) = # OF BYTES TO SEND
;           XSPKMN = # OF PACKETS EXPECTED
;           XSFLG = FLAG BYTE OF 1ST PACKET
;           XSCNT = BYTE COUNT OF 1ST PACKET
;
;           . ***
;           . * SUBSEQUENT XSFLGS
;           . >
;           . * AND XSCNTS
;           . ***
; --

```

.MACRO T)SELF ?A

.NLIST
.LIST ME
.LIST

```

A:  MOV    &TRBUF,R0      ;FORM COMMAND PACKET
     MOVB  &RSCMND,&R0   ;COMMAND FLAG
     MOVB  &RMSIZ,1(R0)  ;SIZE OF MESSAGE
     MOVB  &RSSSLF,2(R0) ;SELF TEST OPERATION
     CLRB  7(R0)         ;NO MODIFIER.
     CLR   4(R0)         ;NO DRIVE OR SWITCHES
     CLR   6(R0)         ;NO SEQUENCE NUMBER
     CLR   8(R0)         ;NO BYTES
     CLR   10(R0)        ;NO RECORD #
     MOV   &RMSIZ,R1     ;GET SIZE
     TST   (R1)+         ;+2 FOR CHECKSUM
     MOV   &RSSNSZ,SNDCNT(R5) ;SIZE TO SEND
     CALL  CHKSUM        ;FORM CHECKSUM
     MOV   R1,(R0)       ;INSERT INTO PACKET
     MOV   &RSEND,XSFLG(R5) ;EXPECT END.
     MOV   &RNSZ,XSCNT(R5) ;THIS BIG
     MOV   &1,XSPKMN(R5) ;AND 1 PACKET
                       ;SEND
     CALL  RSVP          ;RETURN TO SCHEDULER
     BIT   &BIT3,&R5    ;RETRY?(BAD FLAG)
     BNE  A              ;YES

```

.NLIST
.NLIST ME
.LIST
.ENDM

4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850

```

;***
;THE TEST ID MACRO INTERFACES THE SUPERVISOR'S TEST DISPATCH TO THE
;DIAGNOSTIC'S FORMAT BY IMPLEMENTING CALLS THAT: 1) INITIALIZE THE
;PC OF THE TEST CODE (TSTPC(R5)), 2) ASSIGN THE 1ST DRIVES, 3) RUN
;THE TEST, 4) SWITCH DRIVES AND REINITIALIZE, 5) RUN THE TEST AGAIN.
;--

      .MACRO TSTID  ADDR,?A

      .NLIST
      .LIST ME
      .LIST

      MOV     ADDR,TSTTOP      ;SAVE ADDR OF TEST
      CALL   SETUP            ;INIT UNITS ISTPC
      CALL   SETDR            ;GET 1ST DRVS.
      CALL   RUN              ;DO TEST
      CALL   SWAPDR          ;GET NEXT DRVS.
      BCC   A                ;BR NO 2ND DRVS
      CALL   SETUP            ;REINIT UNITS TSTPC
      CALL   RUN              ;REPEAT TEST
                               ;DONE

      A:

      .NLIST
      .LIST ME
      .LIST
      .ENDM

```

```

4853          .SBTTL GLOBAL SUBROUTINES SECTION
4854
4855          ;++
4856          ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES THAT ARE USED
4857          ; TO LINK THE DIAGNOSTIC TO THE SUPERVISOR (THROUGH THE TSTID MACRO).
4858          ;--
4859
4860          ;++
4861          ; SWAPDR
4862          ; SUBROUTINE TO DETERMINE IF TO TEST OTHER DRIVE (FOR ALL UNITS)
4869          ; INPUTS: DR(R5) - DRIVE CONFIGURATION
4870          ;          BLKTB1 - TOP OF DATA BLOCK ALLOCATION TABLE
4871          ;          LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
4872          ;
4873          ; OUTPUTS: DR(R5) UPDATED TO TEST SAME OR OTHER DRIVE
4874          ;          CARRY SET IF SECOND PASS NECESSARY
4917          ;--
4929
4930          SWAPDR:: CLR R2 ;FOR # OF DRIVE 1'S.
4931          005532 005002          MOV #BLKTB1,SWPTR ;TABLE ADDR. OF 1ST UNIT
4932          005534 012737 003352 005650 1$: MOV @SWPTR,R5 ;GET DATA BLOCK ADDR.
4933          005542 017705 000102          BIT #BIT15,R5 ;ABORTED?
4934          005546 032715 100000          BNE 3$ ;YES
4935          005552 001013          BIT #BIT0,DR(R5) ;DID DR. 0?
4936          005554 032765 000001 000060  BNE 3$ ;NO, DID DR.1 1ST PASS
4937          005556 001007          BIT #BIT9,DR(R5) ;YES; 1 SELECTED?
4938          005564 032765 001007 000060  BEQ 3$ ;NO, ALL DONE
4939          005572 001403          INCB DR(R5) ;YES, SWAP
4940          005574 105265 000060          INC R2 ;ONE MORE TO TEST
4941          005602 023727 005650 003370  3$: CMP SWPTR,#LSTDEV ;LAST DEVICE?
4942          005610 103004          BHS 4$ ;YES
4943          005612 062737 000002 005650  ADD #2,SWPTR ;NO-POINT NEXT
4944          005620 000750          BR 1$ ;DO
4945
4946          005622 005702          4$: TST R2 ;(CLEAR CARRY),MORE TO DO?
4947          005624 001410          BEQ 5$ ;NO
4948          005626 022737 020050 003330  CMP #TST3,TSTTOP ;IN TEST 3?
4949          005634 001003          BNE 6$ ;IF NOT, SET CARRY & RETURN
4950          005636 005737 002226          TST DOT3FL ;TEST3-DRIVE 0 ONLY FLAG SET?
4951          005642 001001          BNE 5$ ;IF SET, RETURN WITH CARRY CLEAR
4952          005644 000261          6$: SEC ;SET CARRY TO TEST OTHER DRIVES
4953          005646 000207          5$: RETURN ;RETURN
4954
4955          005650 000000          SWPTR: .WORD
    
```

```

4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969 005652 012737 003352 005726
4970 005660 017705 000042
4971 005664 105065 000060
4972 005670 032765 000400 000060
4973 005676 001002
4974 005700 105265 000060
4975 005704 023727 005726 003370
4976 005712 103004
4977 005714 062737 000002 005726
4978 005722 000756
4979 005724 000207
4980 005726 000000

***
; SETDR - SUBROUTINE TO GET DRIVE FOR 1ST PASS FOR EACH TEST
;
; INPUTS:      DR(R5) - DRIVE CONFIGURATION
;              BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
;              LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
;
; OUTPUTS:     DR(R5) IS SET TO TEST DRIVE 0 OR DRIVE 1
;
***
SETDR:: MOV     #BLKTBL,SETPTR ;TABLE OF ADDR. 1ST UNIT
1:      MOV     @SETPTR,R5    ;GET DATA BLOCK ADDR.
        CLRD   DR(R5)        ;PRESET AS DRO
        BIT    @BIT8,DR(R5)  ;DO DRO?
        BNE   2:             ;YES
        INCB  DR(R5)         ;NO-USE DRIVE 1
2:      CMP    SETPTR,#LSTDEV ;MORE UNITS
        BHIS  3:             ;NO-EXIT
        ADD   @2,SETPTR     ;YES-GET TABLE ENTRY
        BR   1:             ;CONFIGURE THAT UNIT
3:      RETURN
SETPTR: ,WORD

```



```

4983      ;**
4984      ; CLRALL - CLEARS INPUT BUFFER FOR RESPONSE FROM UNIT.
4985      ;
4986      ; INPUTS:      BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
4987      ;              LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
4988      ;
4989      ; OUTPUTS:     ALL UNITS BUFFERS CLEARED.
4990      ;
4991      ; CALLS:       CLRBUF
4992      ;**
4993
4994 005730 012737 003352 006022 CLRALL:: MOV    @BLKTBL,CLRPTR ;TOP OF TABLE OF ADDRESSES
4995 005736 017705 000060 1#      MOV    @CLRPTR,R5 ;GET DATA BLOCK
4996 005742 004737 005770          CALL   CLRBUF ;CLEAR IT'S RECEIVE BUFFER
4997 005746 023727 006022 003370          CMP    CLRPTR,@LSTDEV ;LAST DEV?
4998 005754 103004          BHS    2# ;YES
4999 005756 062737 000002 006022          ADD    @2,CLRPTR ;-->NEXT
5000 005764 000764          BR     1# ;CONTINUE
5001 005766 000207          2#:   RETURN

```

```

5004
5005
5006
5007
5008
5009
5010
5011
5012 005770 CLRBUF:: PUSH R0 ;SAVE R0
      (1) 005770 010046 MOV R0,-(SP)
      (1)
      (1)
5013 005772 PUSH R4 ;SAVE R4
      (1) 005772 010446 MOV R4,-(SP)
      (1)
      (1)
5014 005774 016500 00010: MOV RCVBUF(R5),R0 ;GET ADDRESS OF BUFFER
5015 006000 012704 001036 MOV #RCBFSZ,R4 ;SIZE IN BYTES
5016 006004 005020 1#: CLR (R0)+ ;CLEAR IT
5017 006006 162704 000002 SUB #2,R4 ;2 BYTES LESS
5018 006012 001374 BNE 1# ;MORE
5019 006014 POP R4 ;RESTORE
      (1) 006014 012604 MOV (SP)+,R4
      (1)
5020 006016 POP R0 ;
      (1) 006016 012600 MOV (SP)+,R0
      (1)
5021 006020 000007 RETURN ;EXIT
5022 006022 000000 CLRPTR: .WORD

```

```

5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036 006024 005037 003324
5037 006030 012737 003352 003326
5038 006036 017705 175264
5039 006042 013765 003330 000020
5040 006050 023727 003326 003370
5041 006056 103004
5042 006060 062737 000002 003326
5043 006066 000763
5044 006070 000207

; **
; SETUP - CALLED WITHIN EACH TEST TO INSERT BEGINNING ADDRESS OF THE
; TEST INTO ALL UNITS TEST PC'S.
; INPUTS: TSTTOP LOADED WITH TEST ALGORITHMS STARTING ADDR.
; BLKTB L - TOP OF DATA BLOCK ALLOCATION TABLE
; LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
; OUTPUTS: TSTPC(R5) FOR ALL UNITS
; DONE - CLEARED
; --
SETUP:: CLR DONE ;NOT DONE YET
MOV #BLKTB L, IDPTR ;TABLE TOP ADDR
1$: MOV @IDPTR, R5 ;DEVICE'S DATA BLOCK
MOV TSTTOP, TSTPC(R5) ;INSERT PC FOR TOP OF TEST
CMP IDPTR, #LSTDEV ;ALL UNITS SET?
BHS 2$ ;YES
ADD #2, IDPTR ;NO, GET NEXT POINTER
BR 1$ ;SET HIM UP
2$: RETURN ;DONE

```

```

5047
5048
5049
5050
5051
5052
5053
5054 006072 004737 006122
5055
5056 006076 005737 003324
5057 006102 001006
5058 006104 004737 007172
5059
5060 006110
(3) 006110 104422
5061
5062 006112 004737 010616
5063 006116 000785
5064 006120 000207

```

```

;+
; RUN - IMPLEMENTS THE CALLS TO SEND PACKETS, RECEIVE PACKETS, THEN
; CHECK ANSWERS DURING TEST RUN TIME.
; INPUTS: DONE
; OUTPUTS: NONE
;--
RUN:: CALL NXTST ;MAKE AND SEND NEXT PACK TO ALL
;UNABORTED UNITS
TST DONE ;COMPLETE?
BNE 2$ ;YES
CALL GETANS ;NO,GET ALL RESPONSES
BREAK ;SUPERVISOR CHECK
TRAP C$BRK
CALL CHKANS ;CHECK ALL RESPONSES
BR RUN ;CONTINUE TILL DONE
2$: RETURN

```

```

5067 .SBTTL NXTST / THE SCHEDULER
5068
5069 ;**
5070 ; NXTST - DISPATCH EXECUTION USING EACH UN-ABORTED UNIT'S TEST PROGRAM
5071 ; COUNTER, (TSTPC(R5)). (THE POINTER TO THE TEST CODE THAT COMPRISES
5072 ; MAKING A PACKET AND SENDING IT. CHECKS FIRST FOR ANY UN-ABORTED UNIT
5073 ; THAT IS RETRYING EITHER A DATA ERROR OR A 'INDECIPHERABLE FLAG B/TE'
5074 ; ERROR, IN ORDER TO SERVICE ONLY THAT UNIT THIS PASS. INITIS
5075 ; NON-RETRYING UNITS IF NECESSARY. IF NO RETRIES,DISPATCH ALL
5076 ; UNITS IN ROUND ROBIN FASHION.
5077
5078 ; INPUTS: (IMPLIED) DATA BLOCKS.
5079 ; BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
5080 ; LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
5081
5082 ; OUTPUTS: ERRSF IF ALL UNITS ARE ABORTED.(TO NOTIFY APT)
5083 ; SYSTAT IS UPDATED
5084 ;--
5085
5086 NXTST:: NOP
5087 006122 000240 003352 003314 MOV #BLKTBL,DEVPTR ;UNIT 0 TO START
5088 006174 012737 175156 1$: MCV #DEVPTR,R5 ;GET DATA BLOCK
5089 006132 017707 005715 TST #R5 ;ABORTED?
5090 006136 005715 BMI 2$ ; YES... CHECK NEXT UNIT
5091 006140 100500 000010 3$: BIT #BIT3,#R5 ;NO-RETRY 'BAD FLAG'?
5092 006142 032715 BNE 5$ ;YES...(SEND BREAK,THEN CMD PACK)
5093 006146 001040 BIT #BIT13,#R5 ;NO-RETRYING STILL (NO END PACK YET)?
5094 006150 032715 BEQ 7$ ;NO...
5095 006154 001420 BIT #BIT8,#R5 ;RETRYING A WRITE?
5096 006156 032715 BEQ 4$ ;NO...
5097 006162 001453 SWAPIN ;YES-GET DEVICE REGESTERS
5098 (1) 006164 016500 000006 MOV 6.(R5),R0
5099 (1) 006170 016507 000010 MOV 8.(R5),R1
5100 (1) 006174 016509 000012 MOV 10.(R5),R2
5101 (1) 006200 016505 000014 MOV 12.(R5),R3
5102 (1) 006204 016504 000016 MOV 14.(R5),R4
5103
5104 006210 020205 000206 CMP R2,SAVCNT(R5) ;CURRENT COUNT = SAVED COUNT? (WHERE WE STARTED)
5105 006214 001050 BNE 4$ ;NO...(CONTINUE SENDING DATA PACKS)
5106 006216 042737 000004 003*10 BIC #BIT2,SYSTAT ;YES-CLEAR RETRY FLAGS
5107 006224 042715 020000 BIC #BIT13,#R5
5108 006230 000450 BR 2$ ;CHECK NEXT UNIT.
5109 006232 032715 000000 7$: BIT #BIT10,#R5 ;NO-RETRY DATA ERROR?
5110 006236 001440 BEQ 2$ ;NO...ON TO NEXT UNIT
5111 006240 052737 000002 003310 BIS #BIT1,SYSTAT ;SET RETRY STATUS TO 'DATA ERROR' TYPE
5112 006246 000420 BR 6$ ;YES...
5113
5114 006250 5$: SWAPIN ;GET DEVICE REGISTERS
5115 (1) 006250 016500 000006 MOV 6.(R5),R0
5116 (1) 006254 016507 000010 MOV 8.(R5),R1
5117 (1) 006260 016509 000012 MOV 10.(R5),R2
5118 (1) 006264 016505 000014 MOV 12.(R5),R3
5119 (1) 006270 016504 000016 MOV 14.(R5),R4
5120
5121 006274 010265 000206 MOV R2,SAVCNT(R5) ;SAVE THE BYTE COUNT (FOR WRITE OPERATION)
5122 5110 ;TO MARK HOW MANY DATA PACKS TO SEND
    
```

5111	006300	004737	014030		CALL	DOBRK	;SEND INIT		
5112	006304	032715	100000		BIT	#BIT15,R5	;ABORTED?		
5113	006310	001020			BNE	2#	;YES...		
5114	006312	052737	000004	003310	4#:	BIS	#BIT2,SYSTAT	;NOT ABORTED-SET RETRY STATUS	
5115	006320				6#:	SWAPIN	;GET DEVICE REGISTERS		
(1)	006320	016500	000006				MOV	6.(R5),R0	
(1)	006324	016501	000010				MOV	8.(R5),R1	
(1)	006330	016502	000012				MOV	10.(R5),R2	
(1)	006334	016503	000014				MOV	12.(R5),R3	
(1)	006340	016504	000016				MOV	14.(R5),R4	
(1)									
5116	006344	004775	000020		JSR	PC,@STSTPC(R5)	;DO TEST FOR		
5117	006350	000477			BR	NXTRET	;THIS UNIT ONLY-EXIT		
5118	006352	023727	003314	003370	2#:	CMP	DEVPTR,@LSTDEV	;TRY NEXT UNIT?	
5119	006360	103004			BHIS	NXTST2	;NO		
5120	006362	062737	000002	003314	ADD	#2.,DEVPTR	;YES,->NEXT		
5121	006370	000680			BR	1#	;GET BLOCK		
5122									
5123	006372	005037	006552		NXTST2:	CLR	ABONM	;HERE=NO RETRIES TO DO, NO UNIT ABORTED YET	
5124	006376	012737	003352	003314	MOV	#BLKTBLL,DEVPTR	;-->UNIT 0 STORAGE BLOCK		
5125	006404	017705	174704		PERDEV:	MOV	@DEVPTR,R5	;R5-->NEXT DEVICE STORAGE BLOCK	
5126									
5127	006410	005715			3#:	TST	R5	;ABORTED?	
5128	006412	100426				BMI	4#	;YES	
5129	006414	032715	040000			BIT	#BIT14,R5	;SEND BREAK?	
5130	006420	001407				BEQ	6#	;NO	
5131	006422	004737	014030			CALL	DOBRK	;YES	
5132	006426	032715	040000			BIT	#BIT14,R5	;SUCCESSFUL INIT?	
5133	006432	001016				BNE	4#	;NO ON TO NEXT UNIT	
5134	006434	005715				TST	R5	;ABORTED?	
5135	006436	100414				BMI	4#	;YES-ON TO NEXT UNIT	
5136	006440				6#:	SWAPIN	;NO,GET DEVICE REGISTERS R0-R4 CONTAINING TEST P		
(1)	006440	016500	000006				MOV	6.(R5),R0	
(1)	006444	016501	000010				MOV	8.(R5),R1	
(1)	006450	016502	000012				MOV	10.(R5),R2	
(1)	006454	016503	000014				MOV	12.(R5),R3	
(1)	006460	016504	000016				MOV	14.(R5),R4	
(1)									
5137	006464	004775	000020		JSR	PC,@STSTPC(R5)	;INITIATE 1 PACKET TRANSMISSION AND RETURN		
5138	006470	005715			4#:	TST	R5	;ABORTED?	
5139	006472	100002				BPL	8#	;NO-ON TO NEXT UNIT	
5140	006474	005237	006552			INC	ABONM	;YES...ONE MORE TALLIED	
5141	006500	023727	003314	003370	8#:	CMP	DEVPTR,@LSTDEV	;ALL TU'S TRIED?	
5142	006506	103004				BHIS	5#	;YES	
5143	006510	062737	000002	003314	ADD	#2.,DEVPTR	;NO THE ADDRESS+2=NEXT ADDRESS		
5144	006516	000732			BR	PERDEV	;DO NEXT UNIT		
5145	006520	023737	000010	006552	5#:	CMP	#8.,ABONM	;ALL ABORTED?	
5146	006526	001010				BNE	NXTRET	;NO	
5147	006530					ERRSF	100.,NOMOR	;YES!	
(4)	006530	104454							TRAP C#ERSF
(5)	006532	000144							.WORD 100
(5)	006534	006554							.WORD NOMOR
(5)	006536	000000							.WORD 0
5148	006540				11#:	BREAK	;SUPERVISOR BREAK		
(3)	006540	104420							TRAP C#BRK
5149	006542	005237	003342		INC	ALLGON	;SET DON'T-PRINT STATISTICS FLAG		

5157
5158
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190

.SBTTL RSVP / XOFF AND SEND A PACKET TO ALL DEVICES

```

***
; RSVP - SAVES TEST CODE PROGRAM COUNTER IN TSTPC(R5) AND UNIT'S REGIS-
; TERS. IF NOT IN TEST 8, POINTS TO "XOFF" THAT PRECEEDS PACKET IN
; XMIT BUFFER AND SENDS PACKET WITH XOFF. RETURNS TO SCHEDULER (NXTST)
; SO THAT OTHER UNITS PACKETS MAY BE FORMED, TO GET ALL UNITS WORKING
; AT ONCE. IF IN TEST 8 AND THE UNIT IS NOT MODIFIED, SKIP REST OF
; ROUTINE. IF IN TEST 8 AND THE UNIT IS MODIFIED DO NOT SEND XOFF AND
; PROCEED NORMALLY.
; INPUTS: (SP) CONTAINS UNITS PC TO SAVE SINCE RSVP WAS CALLED. THE
; NUMBER PACKETS EXPECTED (XSPKMN), AND THE EXPECTED FLAGS AND
; BYTE COUNTS OF EACH (XSFLG, XSCNT...) ARE LOADED BY TEST CODE
; (MACROS).
; SNDCNT - # BYTES TO SEND
; REC(R5) - RECORD #
; TRBUF - BUFFER ADDR.
; XSPKMN(R5) - # EXPECTED
; RCVBUF(R5)
;
; OUTPUTS: CMDSNT - UPDATED WITH PACKET OP CODE
; BLKER - RECORD NUMBER STATISTICS UPDATED IF NOT RETRYING
; AND COMMAND PACKET SENT.
; SUCCS(R5) - PRESET CLEAR
; STATUS WORD BR5 - BIT9 - DATA CHECK ERROR - CLEARED
; BIT5 - "VERIFY" OPERATION
; BIT4 - 0 = DATA PACK 1 = CMND
; BIT8 - RD/WR OPERATION
; XSPTR - POINTS TO EXPECTED FLAG
; UPPER BYTE OF XSPKMN IS REPLICATED.
; PACKET POINTER (PKPTR(R5)) POINTS TO TOP OF UNITS RECEIVE BUFFER
; AREA (RCVBUF(R5)) FOR CURRENT UNIT.
---
```

```

5191 006600 000240
5192 006602 010665 000020
5193 006606 010065 000006
(1) 006612 010165 000010
(1) 006616 010265 000012
(1) 006622 010365 000014
(1) 006626 010465 000016
(1)
5194
5195
5196 006632 022737 000002 003344
5197 006640 001007
5198 006642 022765 000000 000210
5199 006650 001523
5200 006652 012700 027746
5201 006656 000404
5202 006660 012700 027745
5203 006664 005265 000070
5204 006670 004737 007122
5205 006674 005715
5206 006676 100510
```

```

RSVP:: NOP ;FINISH TEST
MOV (SP)+,TSTPC(R5) ;SAVE WHERE YOU WERE IN TEST BODY AND
SWAPOW ;SAVE TEST REGISTERS
MOV R0,6.(R5)
MOV R1,8.(R5)
MOV R2,10.(R5)
MOV R3,12.(R5)
MOV R4,14.(R5)
;CORRECT FOR RETURN TO SCHEDULER
;***** IS THIS TEST 9
;***** NO
;***** IF SO, IS THIS UNIT MODIFIED
;***** YES
;FOR NORMAL PACKET SEND
;SEND XOFF+PACKET
;POINT TO XOFF
;ONE MORE TO SEND, TOO.
;SEND BYTE
;R5--> TO STATUS BLK
;ABORTED? YES...QUIT
NOXOFF: MOV #TRBUF,RO
BR SND
XFNSND: MOV #TRBUF-1,RO
INC SNDCNT(R5)
SND: CALL SNDBYT
TST BR5
BMI 6$
```



```

5207 006700 005365 000070          DEC      SNDCNT(R5)      ;NO, SEND MORE
5208 006704 001371          BNE      SND            ;IF MORE TO SEND
5209 006706 012700 027746          MOV      *TRBUF,R0     ;-->BUFFER
5210 006712 016537 000064 003334  MOV      REC(R5),BLKER ;PREPARE FOR RECEIVE
5211 006720 156565 000032 000033  MOV      XSPKMM(R5),XSPKMM+1(R5) ;REPLICATE LO. BYTE TO HI FOR G1PAKS, C
5212 006726 005065 000078          CLR      SUCCS(R5)     ;NO SUCCESS YET
5213 006732 042715 001000          BIC      *BIT9,*R5     ;NO DATA CHK ERROR YET
5214 006736 016565 000102 000104  MOV      RCVBUF(R5),PKPTR(R5) ;TOP OF RCV BUF' .R GOES THE 1ST PACKET
5215 006744 012704 000034          MOV      *XSFLG,R4    ;FORM
5216 006750 060504          ADD      R5,R4        ;ADDRESS
5217 006752 010465 000106          MOV      R4,XSPTR(R5) ;OF 1ST XSFLG
5218
5219 006756 042715 000020          BIC      *BIT4,*R5     ;PRESET AS DATA PAK
5220 006762 121027 000002          CMPB    *R0,*R5CMND   ;WAS IT COMMAND PAK?
5221 006766 001054          BNE      6$           ;NO...
5222 006770 116065 000002 000100  MOVB    2(R0),CMDSNT(R5) ;YES-SAVE COMMAND
5223 006776 052715 000020          BIS      *BIT4,*R5     ;ITS CMND PAK
5224
5225 007002 032715 002000          BIT     *BIT10,*R5    ;RETRYING?
5226 007006 001044          BNE      6$           ;YES-DON'T UPDATE ANY STATS OR CONDITION
5227 007010 126027 000002 000002  CMPB    2(R0),*RSSRD  ;NO,A READ?
5228 007016 001012          BNE      4$           ;NO
5229 007020 042715 000400          BIC      *BIT8,*R5    ;(FOR HARD/SOFT LOGGING) RD/WR FLAG=0
5230 007024 004737 013660          CALL    WHCHDR        ;GET DRIVE
5231 007030 103403          BCS     8$            ;
5232 007032 005265 000114          INC     RDNO(R5)      ;DRIVE 0
5233 007036 000402          BR      4$            ;
5234 007040 005255 000116          8$:    INC     RDN1(R5) ;DRIVE 1
5235
5236 007044 126027 000002 000003  4$:    CMPB    2(R0),*RSSWR ;A WRITE?
5237 007052 001022          BNE      6$           ;NO
5238 007054 052715 000400          BIS      *BIT8,*R5    ;YES, RD/WR FLAG=1
5239 007060 105760 000003          TSTB    3(R0)        ;VERIFY TOO?
5240 007064 001403          BEQ     21$          ;NO
5241 007066 052715 000040          BIS      *BIT5,*R5    ;YES-SET VERIFY FLAG
5242 007072 000402          BR      22$          ;
5243 007074 042715 000040          21$:   BIC      *BIT5,*R5    ;(NO)-RESET VERIFY FLAG
5244 007100 004737 013660          22$:   CALL    WHCHDR        ;GET DRIVE NO
5245 007104 103403          BCS     5$            ;CARRY=DR1
5246 007106 005265 000110          INC     WR(N0(R5))    ;# BLKS WRITTEN DRO
5247 007112 000402          BR      6$            ;EXIT
5248
5249 007114 005265 000112          5$:    INC     WRTN1(R5) ;# BLKS WRITTEN DRV1
5250 007120          6$:
5251 007120 000207          ENDRSP: RETURN      ;RETURN

```

```

5254 .S6TTL SNDBYT / OUTPUT A BYTE TO UNIT
5255
5256 ;**
5257 ; SNDBYT - TEST 'READY' ON INTERFACE. IF 'READY', SEND BYTE AND EXIT.
5258 ; IF TIMED OUT, LOG ERROR.
5259 ; INPUTS - RC = POINTER TO BUFFER
5260 ; - IMPLIED UNIT DATA BLOCK
5261 ; - CSNRDY - TIMEOUT CONSTANT
5262 ; OUTPUTS - RO IS INCREMENTED.
5263 ; ERROR - NOT-READY-TO-SEND TIME OUT
5264 ;--
5265
5266 007122 SNDBYT:: PUSH R1 ;ENTER RO-->BYTE
(1) 007122 010146 MOV R1,-(SP)
(1)
(1)
5267 007124 013701 003346 4$: MOV CSNRDY,R1 ;GET TIMEOUT CONSTANT FOR NOT READY ERROR
5268 007130 105775 000026 1$: TSTB @XMSR(R5) ;READY TO SEND?
5269 007134 100412 BMT 2$ ;YES
5270 007136 010046 PUSH RO ;NO, SAVE RO
(1) 007136 010046 MOV RO,-(SP)
(1)
(1)
5271 007140 BREAK ;MONITOR BREAK
(3) 007140 104422 TRAP C$BRK
5272 007142 POP RO ;RESTORE
(1) 007142 012600 MOV (SP)+,RO
5273
5274 007144 005301 DEC R1 ;ABORTED?
5275 007146 001370 BNE 1$ ;NO
5276 007150 012704 000056 MOV @TOSNDB,R4 ;YES,SET CODE FOR TIMEOUT ERROR
5277 007154 004737 012654 CALL LOG ;LOG IT
5278 007160 000402 BR 3$ ;QUIT
5279 007162 112075 000030 2$: MOVB (RO)+,@XMDB(R5) ;SEND IT
5280 007166 012601 3$: POP R1 ;RESTORE
(1) 007166 012601 MOV (SP)+,R1
(1)
5281 007170 000207 RETURN ;DONE
  
```

```

5284 .SBTTL GETANS / GETS RESPONSES ROUND ROBIN USING "XON"
5285
5286
5287 ;
5288 ; GETANS - IF A UNIT IS RETRYING CLEAR HIS RECEIVE BUFFER (CLRBUF) AND GET
5289 ; HIS RESPONSE (GTPKS1), ELSE, CLEAR ALL BUFFERS (CLRALL) AND
5290 ; GET ALL RESPONSES (GTPKS8).
5291 ; INPUTS: SYSTAT - SYSTEM STATUS WORD.
5292 ;
5293 ; OUTPUTS: SERVST = -1 IF NO RETRIES.
5294 ;--
5295 GETANS:: NOP ;1 UNIT IF RETRY; ELSE ALL
5296 BIT #BIT1!BIT2,SYSTAT ;RETRY?
5297 BNE 1$ ;YES
5298 MOV #-1,SERVST ;PRESET NO UNITS SERVICED
5299 CALL CLRALL ;CLEAR ALL INPUT BUFFERS
5300 CALL GTPKS8 ;GET ALL REPLYs
5301 BR 2$ ;EXIT
5302 1$: CALL CLRBUF ;RETRY-CLEAR 1 UNIT ONLY
5303 ;RS->UNIT BY NXTST
5304 CALL GTPKS1 ;GET 1 REPLY
5305 2$: RETURN ;DONE
5306
5307 GETPTR: .WORD
007172 000240
007174 032737 000006 003310
007202 001010
007204 012737 177777 010362
007212 004737 005730
007216 004737 007450
007222 000404
007224 004737 005770
007230 004737 007240
007234 000207
000000

```

5310
 5311
 5312
 5313
 5314
 5315
 5316
 5317
 5318
 5319
 5320
 5321
 5322
 5323
 5324 007240 000240
 5325 007242 012703 000034
 5326 007246 060503
 5327 007250 010301
 5328 007252 062701 000002
 5329 007256 012700 007446
 5330 007262 004737 007122
 5331
 5332 007266 016500 000102
 5333 007272 116502 000033
 5334 007276 032702 177400
 5335 007302 011137 003320
 5336 007306 011337 003316
 5337 007312 004737 010366
 5338 007316 032715 100000
 5339 007322 001050
 5340 007324 005300
 5341 007326 111037 003311
 5342 007332 121037 003316
 5343 007336 001420
 5344 007340 121027 000002
 5345 007344 001006
 5346 007346 012737 000016 003320
 5347 007354 012702 000001
 5348 007360 000407
 5349 007362 121027 000001
 5350 007366 001026
 5351 007370 012737 000204 003320
 5352 007376 005202
 5353
 5354 007400 005200
 5355 007402 005337 003320
 5356 007406 001411
 5357 007410 004737 010366
 5358 007414 005765 000074
 5359 007420 001011
 5360 007422 032715 100000
 5361 007426 001006
 5362 007430 000764
 5363
 5364 007432 005302
 5365 007434 001403

```
.SBTTL GTPKS1 / GET RETRY RESPONSE-1 UNIT
;
; GTPKS1 - SENDS 'XON' TO UNIT, GETS FLAG BYTE (IF ANY), CHECKS IF IT IS
; WHAT WAS EXPECTED. IF IT IS, USE EXPECTED BYTE COUNT(XSCNT), IF
; NOT, CHECK IF PREMATURE-END PACK OR (SINCE MAINTENANCE MODE)
; IF IT'S A PREMATURE DATA PACK. ADJUST COUNT, GET REST OF
; PACKET, AND REPEAT ABOVE UNTIL NO MORE PACKETS.
; INPUTS: (IMPLIED) UNITS DATA BLOCK
;          RSND$ - END PACKET SIZE
; OUTPUTS: SYSTAT UPPER BYTE = FLAG BYTE RECEIVED
;
GTPKS1:: NOP ;R5->THE UNIT
MOV        #X$FLG,R3 ;THE OFFSET VALUE OF FLAG
ADD        R5,R3      ;FORM THE ABSOLUTE ADDRESS
MOV        R3,R1      ;R3-->ADDR. OF EXPECTED FLAG
ADD        #2,R1      ;R1-->ADDR. OF EXPECTED COUNT
MOV        #EXON,R0   ;R0=ADDRESS
CALL       SNDBYT     ;XON THE DEVICE
;*** TIME CRITICAL
MOV        RCVBUF(R5),R0 ;***--> TO THE BUFFER
MOVB      X$PKNM+1(R5),R2 ;***GET THE # OF PACKETS TO RECEIVE
BIT        #177400,R2  ;***SIGN UN-EXTEND
1$: MOV     #R1,RCBCNT  ;***HOW MANY BYTES IT SHOULD BE
MOV        #R3,RCFLG  ;***WHAT THE FIRST BYTE SHOULD BE
CALL       GTBYTE     ;***GET THE ALL IMPORTANT FLAG
BIT        #BIT15,#R5 ;TIMEOUT?
BNE        4$         ;YES
DEC        R0         ;-> BYTE RECIEVED
MOVB      #R0,SYSTAT+1 ;SAVE IT AS FLAG BYTE
CMPB      #R0,RCFLG  ;1ST BYTE WHAT WAS EXPECTED?
BEQ        2$         ;YES
CMPB      #R0,#RSEND ;NO, WAS IT END PAK?
BNE        14$        ;NO
MOV        #RSND$Z,RCBCNT ;YES, USE END SIZE FOR COUNT
MOV        #1,R2      ;AND ASSUME IT'S LAST PACKET?
BR         2$         ;CONTINUE RECEIVE
14$: CMPB  #R0,#RSDATA ;WAS IT DATA?
BNE        4$         ;NO,CHKANS MAY FIND INIT...
MOV        #RSDASZ,RCBCNT ;YES, SET FOR DATA PAK SIZE
INC        R2         ;ONE MORE PACK THAN EXPECTED (END PAK)
2$: INC    R0         ;RESTORE TO -> NEXT BYTE
5$: DEC    RCBCNT     ;THAT'S ONE LESS BYTE TO GO
BEQ        3$         ;DONE
CALL       GTBYTE     ;GET REST OF PACKET
TST       DLV(R5)    ;ERROR
BNE        4$         ;YES-ALL OVER
BIT        #BIT15,#R5 ;OR IF ABORTED
BNE        4$         ;THEN QUIT
BR         5$         ;CONTINUE RECEIVE
3$: DEC    2         ;ONE LESS PACKET TO GO
BEQ        4$         ;MORE PACKETS IN TRANSACTION?
```

B5

GLOBAL AREAS MACY11 30(1046) 25-JAN-84 08:33 PAGE 28-1
CZTUUF.P11 25-JAN-84 08:09 GTPKS1 / GET RETRY RESPONSE-1 UNIT

SEQ 0053

5366
5367 007436 022121
5368 007440 022323
5369 007442 000717
5370 007444 000207
5371
5372 007446 020
5373 007447 023

CMP (R1)+,(R1)+
CMP (R3)+,(R3)+
7R 1+
4+: RETURN
EXON: .BYTE RSXON
EXOFF: .BYTE RSXOFF

;YES
;POINT TO NEW EXPECTED COUNT
;AND FLAG,
;AND RECEIVE,
;RETURN

```

5376
5377
5378
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5390
5391
5392
5393
5394
5395 007450 000240
5396 007452 022737 000002 003344
5397 007460 001006
5398 007462 022765 000000 000210
5399 007470 001002
5400 007472 000137 010166
5401 007476 012737 003352 010364
5402 007504 017705 000654
5403 007510 012715 100000
5404 007514 001404
5405 007516 004737 010276
5406 007522 000137 010124
5407 007526 105765 000033
5408 007532 001004
5409 007534 004737 010276
5410 007540 000137 010124
5411 007544 105365 000033
5412 007550 017537 000106 003316
5413 007556 062765 000002 000106
5414 007564 017537 000106 003320
5415 007572 022737 000002 003344
5416 007600 001404
5417 007602 012700 007446
5418
5419 007606 004737 007122
5420 007612 016500 000104
5421 007616 004737 010366
5422 007622 032715 100000
5423 007626 001404
5424 007630 105065 000033
5425 007634 000137 010124
5426 007640 005300
5427 007642 111037 003311
5428 007646 121037 003316
5429 007652 001436
5430 007654 105065 000033
5431 007660 121027 000002

```

```

.SBTTL GTPKS8 / GET RESPONSES (NO RETRIES)
***
; GTPKS8 - IF IN TEST 9 AND THE UNIT IS NOT MODIFIED, SKIP THE REST
; OF THE ROUTINE. OTHERWISE,
; SET ALL ABORTED UNITS SERVICED (SERVST: BIT POSITION). UNTIL
; ALL UNITS SERVICED (SERVST=0). IF NO MORE PACKETS, SET UNIT
; SERVICED. ELSE, GET A FLAG BYTE FROM UNIT, DECREMENTING THE
; NUMBER OF PACKETS LEFT. CHECK TO SEE IF EXPECTED FLAG,
; ADJUST COUNT IF NOT, GET REST OF PACKET. IF WAS DATA PACK,
; AND NOT IN TEST 9, SEND "XOFF" TO ENHANCE THROUGHPUT AND GO ON
; TO NEXT UNIT (IF ANY). IF IN TEST 9, DO NOT SEND "XOFF".
; INPUTS: (IMPLIED) UNITS DATA BLOCK POINTED TO BY R5. NONE PASSED.
; R5NSZ - END PACK SIZE
; R5DNSZ - DATA + END SIZE
;
; OUTPUTS: SYSTAT - UPPER BYTE=1ST BYTE RECEIVED, CURRENT UNIT
;--
GTPKS8:: NOP ;GET ALL UNITS RESPONSES XOFF IF DATA PAK (THROU
;***** IS THIS TEST 9
;***** NO
;***** IF SO, IS THIS UNIT MODIFIED
;***** YES, CONTINUE NORMALLY
;***** ELSE, SKIP ROUTINE
11: MOV #BLKTB,GTPT ;--1ST
GTAGIN: MOV #GTPT,R5 ;GET DATA BLOCK
;ABORTED?
;NO
;YES-SET 'SERVICED' AND
;ON TO NEXT UNIT
21: TSTB XSPKNM+1(R5) ;NO, ANY PACKETS LEFT?
;YES
;NO-HE'S DONE
;SO ON TO NEXT UNIT
31: DECB XSPKNM+1(R5) ;NOW ITS ONE LESS PACKET
;GET EXPECTED FLAG
;--> COUNT
;AND EXPECTED COUNT
;***** IF TEST 9
;***** DO NOT SEND XON
;--> XON
;***TIME CRITICAL
;***SEND IT
;***-->WHERE 1ST BYTE GOES
11: MOV PKPTR(R5),R0
;***GET IT
;ABORTED?
;NO-CONTINUE
;YES-NO MORE PACKETS EXPECTED
;ON TO NEXT
;-->BYTE JUST RECEIVED
;SAVE IT
;IS IT WHAT EXPECTED?
;YES
41: DEC R0
;BYTE JUST RECEIVED
;SAVE IT
;IS IT WHAT EXPECTED?
;YES
UNXPCT: CLRB XSPKNM+1(R5) ;NO, MUST BE LAST REPLY
;MAYBE AN END PAK?
;***** IF TEST 9
;***** DO NOT SEND XON
;--> XON
;***TIME CRITICAL
;***SEND IT
;***-->WHERE 1ST BYTE GOES
;***GET IT
;ABORTED?
;NO-CONTINUE
;YES-NO MORE PACKETS EXPECTED
;ON TO NEXT
;-->BYTE JUST RECEIVED
;SAVE IT
;IS IT WHAT EXPECTED?
;YES
;NO, MUST BE LAST REPLY
;MAYBE AN END PAK?

```

```

5432 007664 001004          BNE      4#      ;NO
5433 007666 012737 000016 003320  MOV     #RSNSZ,RCBCNT ;YES, USE PROPER COUNT
5434 007674 002406          BR      GTUM      ;AND GET IT
5435 007676 121027 000001      4#:  CMPB   #R0,#RSDATA ;IS IT DATA?
5436 007702 001110          RNE     GTDOWN    ;NO, ALL OVER, CHKANS WILL INIT UNIT
5437 007704 012737 000222 003320  MOV     #RSNSZ,RCBCNT ;YES, USE COUNT OF DATA + END PAK SURE TO FOLLOW
5438 007711 005200          GTUM:  INC     R0      ;WHERE TO STUFF THE REST
5439 007714 005337 0J3320      5#:  DEC     RCBCNT     ;ONE DOWN
5440 007720 001501          BEQ     GTDOWN    ;NONE TO GO
5441 007722 004737 010356      CALL   GTBYTE     ;MORE TO GO
5442 007726 032715 100000      BIT     #BIT15,#R5 ;TIMEOUT?
5443 007732 001074          BNE     GTDOWN    ;YES
5444 007734 005765 000074      TST    DLV(R5)    ;BUT DLV ERROR?
5445 007740 001765          BEQ     5#      ;NO
5446 007742 105065 000033      CLRB   XSPKM+1(R5) ;YES-LAST TIME
5447 007746 000466          BR      GTDOWN    ;ON TO NEXT
5448
5449 007750 005200          GTOK:  INC     R0      ;NEXT PLACE IN BUFFER
5450
5451
5452 007752 022737 000002 003344      1#:  CMP     #2,TEST9 ;*** REV. - IF, NOT TEST 9
5453 007760 001022          BNE     7#      ;*** REV. - THEN, NO MRSP HANDSHAKE REQUIRED
5454 007762          PUSH   R0      ;*** REV. - ELSE, TEST MRSP HANDSHAKE.
(1) 007762 010046          MOV     R0,-(SP)
(1)
5455 007764 012737 000002 010274          MOV     #2,MRSDLY ;*** REV. - DELAY FOR WAIT LOOP
5456
5457 007772 005000          2#:  CLR     R0      ;*** REV. - THIS IS THE BEGINNING DELAY LOOP
5458 007774 005300          3#:  DEC     R0      ;*** REV. -
5459 007776 001376          BNE     3#      ;*** REV. -
5460 010000 005337 010274          DEC     MRSDLY    ;*** REV. -
5461 010004 001372          BNE     2#      ;*** REV. - THIS IS THE END OF DELAY LOOP
5462
5463 010006 105775 000022          TSTB   #RCS(R5)   ;*** REV. - IF, DONE SET,
5464 010012 001066          BNE     ERRMOD    ;*** REV. - THEN, IT'S AN ERROR BECAUSE
5465
5466 010014 012700 010272          MOV     #MODRSP,R0 ;*** REV. - THERE WAS NO MRSP HANDSHAKE.
5467 010020 004737 007122          CALL   SNDBYT     ;*** REV. - ELSE, SEEMS TO BE OK, LETS
5468 010024          POP     R0      ;*** REV. - SEND A 'CONTINUE' AND
(1) 010024 012600          MOV     (SP)+,R0 ;*** REV. - SEE IF HANDSHAKE WORKS.
(1)
5469
5470 010026 005337 003320          7#:  DEC     RCBCNT     ;MORE BYTES?
5471 010032 001413          BEQ     4#      ;NO-ALL DONE
5472 010034 004737 010366          CALL   GTBYTE     ;YES-GET IT
5473 010040 032715 100000      BIT     #BIT15,#R5 ;TIMEOUT?
5474 010044 001027          BNE     GTDOWN    ;YES
5475 010046 005765 000074      TST    DLV(R5)    ;ERROR?
5476 010052 001737          BEQ     1#      ;NO
5477 010054 105065 000033      CLRB   XSPKM+1(R5) ;LAST TIME
5478 010060 000421          BR      GTDOWN    ;EXIT
5479 010062 122775 000001 000104      4#:  CMPB   #RSDATA,#PKPTR(R5) ;WAS DATA?
5480 010070 001015          BNE     GTDOWN    ;NO, ALL DONE
5481 010072 010065 000104          MOV     R0,PKPTR(R5) ;START OF NEXT PACK NEXT TIME
5482

```

5483	010076	022737	000002	003344		CMP	#2,TEST9	;*** REV. - IF, TEST 9	
5484	010104	001003				BNE	20#	;*** REV. - ELSE,	
5485	010106	005765	000210			TST	MRSP(R5)	;*** REV. - ANDIF, MRSP	
5486	010112	001004				BNE	GTDOWN	;*** REV. - THEN, NO HANDSHAKE	
5487								;*****	
5488	010114	012700	007447		20#:	MOV	#EXOFF,RO	;XOFF AND SEND TO	
5489	010120	004737	007122			CALL	SNOBYT	;ENHANCE THROUGHPUT	
5490	010124	062765	000002	000106	GTDOWN:	ADD	#2, XSPTR(R5)	;NEXT XSFLG FOR NEXT TRY	
5491	010132	023727	010364	003370		CMP	GTPTR, #LSTDEV	;DONE ONE CYCLE ALL UNITS?	
5492	010140	103005				BHIS	1#	;YES	
5493	010142	062737	000002	010364		ADD	#2, GTPTR	;NEXT UNIT	
5494	010150	000137	007504			JMP	GTAGIN	;CONTINUE RECEIVE	
5495	010154	105737	010362		1#:	TSTB	SERVST	;DONE SERVICING ALL PAKS	
5496								;FROM ALL UNITS?	
5497	010160	001402				BEG	ENDGP8	;YES	
5498	010162	000137	007450			JMP	GTPKSB	;NO, KEEP TRYING	
5499	010166	000207			ENDGP8:	RETURN		;RETURN	
5500									
5501	010170	000240			ERRMOD:	NOP		;*** REV. - MRSP ERROR	
5502	010172					PRINTF	#MESMRS,UNITNO		
(8)	010172	013746	027412						MOV UNITNO, -
(7)	010176	012746	010220						MOV #MESMRS,
(6)	010202	012746	000002						#2, -(SP)
(3)	010206	010600							MOV SP,RO
(4)	010210	104417							TRAP C#PNTF
(4)	010212	062706	000006						ADD #6,SP
5503	010216	000207				RETURN			
5504									
5505	010220	047045	051445	022471	MESMRS:	.ASCIZ	!#N#S9#S2#01#S9#S9#AERROR IN MRSP PROTOCOL!		
5506						.EVEN			
5507	010272	020			MODRSP:	.BYTE	RSCONT		
5508		010274				.EVEN			
5509	010274	000000			MRSDLY:	.WORD			


```

5512          .SBYTL  SETSRV / SET UNIT SERVICED
5513
5514          ;++
5515          ; SETSRV - RESET THE BIT IN 'SERVST' CORRESPONDING TO THE UNIT NUMBER.
5516          ; INPUTS - SERVST - 'SERVICED' WORD
5517          ;          - SR5 = UNIT # (BITS 0, 1, 2)
5518          ; OUTPUTS - SERVST MODIFIED
5519          ;--
5520
5521 010276      SETSRV: PUSH      R5          ;SET UNIT SERVICED
(1) 010276 010546      MOV        R5,-(SP)
(1)
(1)
5522 010300      PUSH      R0          MOV        R0,-(SP)
(1) 010300 010046
(1)
(1)
5523 010302 011505      MOV        SR5,R5          ;GET STAT WD
5524 010304 042705 177770      BIC        #177770,R5      ;MASK UNIT #
5525 010310 012700 010342      MOV        #SRVTBL,R0      ;-->TOP OF BIT TABLE
5526 010314 005705      1$: TST        R5          ;RIGHT ONE?
5527 010316 001404      BEQ        2$          ;YES
5528 010320 062700 000002      ADD        #2,R0          ;NO, -->NEXT
5529 010324 005305      DEC        R5          ;1 LESS
5530 010326 000772      BR         1$          ;CONTINUE
5531 010330 041037 010362      2$: BIC        SR0,SERVST    ;NOW IT DOWN
5532 010334      POP        R0          MOV        (SP)+,R0
(1) 010334 012600
(1)
5533 010336      POP        R5          MOV        (SP)+,R5
(1) 010336 012605
(1)
5534 010340 000207      RETURN          ;RETURN
5535
5536 010342 000001      SRVTBL: .WORD   BIT0          ;BIT POSITION LOOKUP TABLE
5537 010344 000002      .WORD   BIT1
5538 010346 000004      .WORD   BIT2
5539 010350 000010      .WORD   BIT3
5540 010352 000020      .WORD   BIT4
5541 010354 000040      .WORD   BIT5
5542 010356 000100      .WORD   BIT6
5543 010360 000200      .WORD   BIT7
5544
5545 010362 000000      SERVST: .WORD
5546 010364 000000      GTPTR:  .WORD
  
```

5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572 010366 005037 010612
5573 010372 013704 003350
5574 010376 105775 000022
5575 010402 100013
5576 010404 017565 000024 000074
5577 010412 116520 000074
5578 010416 005765 000074
5579 010422 100472
5580 010424 005065 000074
5581 010430 000467
5582 010432 005337 010612
5583 010436 001357
5584
5585
5586
5587 010440 010037 010614
5588 010444 012700 007447
5589 010450 004737 007122
5590 010454 105775 000022
5591 010460 100415
5592 010462 005337 010612
5593 010466 105737 010612
5594 010472 001370
5595 010474
(3) 010474 104422
5596 010476 012700 007446
5597 010502 004737 007122
5598 010506 013700 010614
5599 010512 000426
5600 010514 013700 010614
5601 010520 017565 000024 000074
5602 010526 116520 000074
5603 010532 005765 000074

.SBTTL GTBYTE / GET A BYTE FROM UNIT

```

;***
; GTBYTE - TEST INTERFACE FOR 'READY-TO-RECEIVE' AND INPUT A BYTE. IF
; SO. IF NOT, THE FOLLOWING OCCURS: SEND 'XOFF' TO UNIT IN
; PREPARATION FOR +C CHECK ('BREAK' TO SUPERVISOR). WAIT
; TO SEE IF A CHARACTER SLOPS OVER DUE TO UART LATENCY. IF
; ONE DOES THEN MIGHT AS WELL GET IT AND SEND 'XON' TO GET
; THE REST OF THE MESSAGE, OTHERWISE, 'BREAK'. THEN SEND
; 'XON', AND TEST FOR LONG TIMEOUT (A 30 SECOND REWIND). IF SO,
; LOG ERROR, OTHERWISE REPEAT THE ABOVE UNTIL READY OR TIME OUT.
; REMEMBER TO PRESERVE R0 SINCE THE 'BREAK' TRAP CLOBBERS IT.

```

```

; INPUTS - R0 POINTS TO INPUT BUFFER
;         - IMPLIED UNITS DATA BLOCK
;         - CSRCVB TIME OUT MULTIPLIER
;
; OUTPUTS - R0 IS INCREMENTED
;         - DLV (R5) NON-ZERO ON INTERFACE ERROR.
;
; ERROR - TIME OUT ON RECEIVE
; --

```

```

GTBYTE:: CLR      GBTMP      ; TIMEOUT REGISTER
        MOV      CSRCVB,R4  ; TIMEOUT ERROR CONSTANT (MULTIPLIER)
1$:     TSTB     SRCR(R5)    ; READY?
        BPL      3$        ; NO
        MOV      SRCDB(R5),DLV(R5) ; GET ERROR + BYTE
        MOVB     DLV(R5),(R0) ; COPY BYTE TO BUFFER
        TST      DLV(R5)    ; ERROR?
        BMI      4$        ; YES-EXIT
        CLR      DLV(R5)    ; NO-RESET
        BR       4$        ; AND EXIT
3$:     DEC      GBTMP      ; DEC T.O. CONSTANT
        BNE      1$        ; STILL VALID

```

```

; CODE TO SEE +C DURING LONG SEEK OR REWIND
; HERE GBTMP=0
; R0 MUST BE PRESERVED!
; QUIET THE DEVICE
; BY SENDING XOFF
6$:     TSTB     SRCR(R5)    ; CHARACTER SLOP OVER?
        BMI      5$        ; YES
        DEC      GBTMP      ; NO-WAIT A WHILE
        TSTB     GBTMP      ; DONE WAITING?
        BNE      6$        ; NO
        BREAK    ; YES-NO SLOP OVER

```

```

; START DEVICE TALKING
; AGAIN
; RESTORE R0
; END KLUGE
5$:     MOV      GBTMP2,R0  ; RESTORE R0
        MOV      SRCDB(R5),DLV(R5) ; GET ERROR + BYTE
        MOVB     DLV(R5),(R0) ; COPY BYTE TO BUFFER
        TST      DLV(R5)    ; ERROR?

```

TRAP C\$BRK

5604	010536	100403		BMI	17#		;YES-EXIT
5605	010540	005065	000074	CLR	DLV(R5)		;NO-CLEAR
5606	010544	000400		BR	17#		;EXIT
5607	010546	010037	010614	17#:	MOV	R0,GBTMP2	;AGAIN SAVE R0
5608	010552	012700	007446		MOV	#EXON,R0	;RESTORE TO TALKING STATE
5609	010556	004737	007122		CALL	SNDBYT	;BY SENDING 'XON'
5610	010562	013700	010614		MOV	GBTMP2,R0	;RESTORE R0
5611	010566	000410			BR	4#	;DONE
5612	010570	005037	010612	7#:	CLR	GBTMP	
5613	010574	005304			DEC	R4	;TIMEOUT?
5614	010576	001277			BNE	1#	;NO
5615	010600	012704	000050		MOV	#TORCVB,R4	;YES
5616	010604	004737	012654		CALL	LOG	;LOG ERROR.
5617	010610	000207		4#:	RETURN		;RETURN
5618	010612	000000		GBTMP:	.WORD	0	
5619	010614	000000		GBTMP2:	.WORD	0	

5622
 5623
 5624
 5625
 5626
 5627
 5628
 5629
 5630
 5631
 5632
 5633
 5634
 5635
 5636
 5637
 5638
 5639
 5640
 5641
 5642
 5643
 5644
 5645
 5646
 5647
 5648
 5649
 5650
 5651
 5652
 5653
 5654
 5655
 5656
 5657
 5658
 5659

010616 000240
 010620 032737 000006 003310
 010626 001403
 010630 004737 010726
 010634 000432
 010636 012737 003357 010724
 010644 017705 000054
 010650 032715 100000
 010654 001012
 010656 022737 000002 003344
 010664 001004
 010666 022765 000000 000210
 010674 001402
 010676 004737 010726
 010702 023727 010724 003370
 010710 103004
 010712 062737 000002 010724
 010720 000751
 010722 000207
 010724 000000

```
.SBTTL  CHKANS / CHECK DEVICE(S) RESPONSE
;***
; CHKANS - AS IN "GETANS", IF RETRYING DO ONLY 1 UNIT ELSE DO ALL NON-
;         ABORTED UNITS. NOTE, IF IN TEST 9 AND THE UNIT IS NOT
;         MODIFIED DO NOT CHECK UNIT.
; INPUTS: IMPLIED SYSTAT BIT1 (RETRYING)
;         BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
;         LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
;
; OUTPUTS: NONE PASSED.
;--
CHKANS:: NOP                                ;IF RETRY THEN CHECK ONE
;ELSE CHECK ALL
;RETRYING?
;NO DO NORMAL
;YES DO SINGLE UNIT
;R5 -> UNIT
;ALL DONE
;
CHK8:  MOV  #BLKTBL,CHKPTR ;YOU KNOW ... TOP OF TABLE
2$:   MOV  @CHKPTR,R5      ;GET UNIT'S BLOCK ADDRESS
      BIT  @BIT15,@R5     ;ABORTED?
      BNE  3$             ;YES
      CMP  #2,TEST9      ;***** IS THIS TEST 9
      BNE  1$             ;***** NO-CONTINUE NORMALLY
      CMP  #0,MRSR(R5)    ;***** IF SO, IS THIS UNIT MODIFIED
      BEQ  3$             ;***** NO SKIP NEXT INSTR
1$:   CALL  CHKPXS        ;NO. DO THIS GUY
3$:   CMP  CHKPTR,@LSTDEV ;ALL DONE?
      BHS  CHKANR        ;YES
      ADD  #2,CHKPTR     ;NO.-->NEXT DEVICE
      BR   2$            ;DO DA
;
CHKANR: RETURN
CHKPTR: .WORD
```

5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717

010726 000240
010730 042715 000010
010734 016500 000102
010740 116502 000032
010744 012703 000034
010750 060503
010752 010301
010754 062701 000002
010760 010065 000104
010764 111037 003311
010770 011137 003320
010774 011337 003316
011000 121013
011002 001057
011004 121027 000020
011010 001534
011012 013704 003320
011016 005744
011020 004737 013770
011024 103005
011026 012704 000022
011032 004737 012654
011036 000521
011040 122710 000002
011044 001005
011046 004737 011322
011052 012702 000001
011056 000511
011060 122710 000001
011064 001012

.SBTTL CHKPKS / DECIPHERS RESPONSE OF UNIT POINTED TO BY R5 /

; CHKPKS - FOR UNIT R5 AND FOR ALL PACKETS, CHECK TO SEE IF PACKET IS DATA OR
; END PACK, CHECK CHECKSUMS, COMPARE DATA IF DATA PACK, CHECK
; SUCCESS CODE IF END. IF UNKNOWN PACKET TYPE, CHECK FOR INTERFACE
; ERROR. IF "CONTINUE" FALL THROUGH. IF "INIT" SET "SEND
; BREAK" FLAG. CALL "LOG" WITH R4=ERROR NUMBER IF ERROR.
; THIS ROUTINE IS ALSO USED TO DETERMINE THE PROTOCOL OF A UNIT. IN
; THE FIRST PART OF TEST 9 A GET CHARACTERISTICS COMMAND PACKET WAS
; SENT TO THE TUS8. IF THE RESPONSE WAS A DATA PACKET, WHICH IS
; EXPECTED, THEN THE UNIT IS NOT MODIFIED, AND THE MRSP FLAG IS
; CLEARED. IF THE RESPONSE IS AN END PACKET, WHICH WOULD BE
; HANDLED BY THIS ROUTINE AS AN UNKNOWN, THEN THE UNIT IS MODIFIED,
; AND THE MRSP FLAG IS SET.
; INPUTS: (IMPLIED) UNITS DATA BLOCK
; OUTPUTS: ERRORS - DLV ERROR
; - UNKNOWN FLAG BYTE ERROR
; - CHECKSUM ERROR
; - DATA COMPARE ERROR
; R4 = ERROR NUMBER
; SYSTAT UPPER BYTE = 1ST BYTE OF RESPONSE
;--

CHKPKS:: NOP ;CHECK WHAT WAS RECIEVED
BIC #8IT3, R5 ;CLEAR 'BAD FLAG' RETRY BIT
MOV RCVBUF(R5), R0 ;GET BUFFER ADDR.
MOVB XSPKNT(R5), R2 ;AND # OF PACKETS EXPECTED
MOV #XSFLG, R5 ;THE OFFSET VALUE
ADD R5, R3 ;R3 -> THIS UNIT XSFLG AGAIN
MOV R3, R1 ;COPY TO R1
ADD #2, R1 ;R1 --> XSBCNT FOR 1ST PACKET
1\$: MOV RO, PKPTR(R5) ;POINT TO PACKET
MOVB #R0, SYSTAT+1 ;SAVE RCV'D BYTE
MOV #R1, RCBCNT ;GET COUNT
MOV #R3, RCFLG ;AND FLAG
CMPB #R0, #R3 ;1ST BYTE=EXPECTED?
BNE 5\$;UH OH...
CMPB #R0, #RSCONT ;OK, IS IT 1 BYTE?
BEQ 7\$;YES... ONTO NEXT PACK
MOV RCBCNT, R4 ;NO, SO > 1 BYTE (NEVER EXPECT INIT!)
TST -(R4) ;EXPECTED, SO COUNT MUST BE RIGHT
CALL CKCKSM ;ADJUST FROM RECEIVE COUNT TO COUNT FOR CHECKSUM
BCC 2\$;CHECK CHECKSUM
MOV #BDCHK, R4 ;NO CARRY... NO INCORRECT
CALL LOG ;ERROR
BR 7\$;LOG IT
2\$: CMPB #RSEND, (R0) ;ON TO NEXT PACK
BNE 3\$;END PAK?
CALL CHKEND ;NO
MOV #1, R2 ;YES-CHECK
BR 7\$;LAST PACKET
3\$: CMPB #RSDATA, #R0 ;AND FALL THROUGH
BNE 4\$;DATA PAK?
 ;NO

```

5718 011066 022737 000001 003344      CMP      #1,TEST9      ;***** IS THIS TEST 9
5719 011074 001003                    BNE      11$           ;***** NO-CONTINUE NORMALLY
5720 011076 005065 000210              CLR      MRSP(R5)     ;***** CLR MRSP FLAG
5721 011102 000402                    BR       12$           ;***** SKIP INSTR
5722 011104 004737 014570              11$:    CALL     COMPAR ;YES-CHECK DATA
5723 011110 000474                    12$:    BR       7$           ;ALL DONE?
5724 011112 052715 020010              4$:    BIS      #BIT3!BIT13,RS5 ;SET 'BAD FLAG' RETRY FLAGS
5725 011116 012704 000010              MOV      #OTL,R4      ;OUT TO LUNCH
5726 011122 005765 000074              TST      DLV(R5)      ;AH,BUT DLV ERROR?
5727 011126 001402                    BEQ      20$           ;NO
5728 011130 012704 000012              MOV      #OVRN,R4     ;YES-USE CORRECT ERROR #
5729 011134 004737 012654              20$:    CALL     LOG      ;TALLY
5730 011140 000467                    BR       8$           ;DONE
5731
5732                                     ;HERE CHECKS UNEXPECTED RESPONSE
5733
5734 011142 122710 000004              5$:    CMPB     #RSINIT,RO ;INIT?
5735 011146 001007                    BNE      6$           ;NO
5736 011150 052715 020010              BIS      #BIT3!BIT13,RS5 ;YES-SET RETRY FLAGS
5737 011154 012704 000006              MOV      #RCINIT,R4   ; WE GOT AN INIT
5738 011160 004737 012654              CALL     LOG          ;TALLY IT
5739 011164 000455                    BR       8$           ;DONE
5740 011166 122710 000001              6$:    CMPB     #RSDATA,RO ;DATA PAK?
5741 011172 001013                    BNE      9$           ;NO
5742 011174 012704 000204              MOV      #RSDASZ,R4   ;YES, USE DATA SIZE
5743 011200 005744                    TST      -(R4)        ;ADJUST FOR CHKSUM
5744 011202 004737 013770              CALL     CKCKSM       ;AND CHECK
5745 011206 103430                    BCS      10$          ;GOOF
5746 011210 004737 014570              CALL     COMPAR       ;OK, HOW'S THE DATA?
5747                                     ;EXPECTED END, GOT
5748                                     ;DATA + END.
5749 011214 062700 000204              ADD      #RSDASZ,RO   ;POINT TO END PACK
5750 011220 000657                    BR       1$           ;CHECK IT, USE SAME XSFLG

```

5752	011222	122710	000002		9\$:	CMPB	#RSEND,(R0)	;END?
5753	011226	001331				BNE	4\$;NO-OUT TO LUNCH
5754	011230	012704	000016			MOV	#RSSNSZ,R4	;YES, TOTAL SIZE MINUS
5755	011234	005744				TST	-(R4)	;TWO (THE CHKSUM)
5756	011236	004737	013770			CALL	CKCKSM	;CHECK IT
5757	011242	103412				BCS	10\$;OOPS
5758	011244	022737	000001	003344		CMP	#1,TEST9	;***** IS THIS TEST 9
5759	011252	001003				BNE	13\$;***** NO-CONTINUE NORMALLY
5760	011254	012765	000001	000210		MOV	#1,MRSP(R5)	;***** IF SO, SET THE MRSP FLAG
5761	011262	004737	011322		13\$:	CALL	CHKEND	;OK,NOW TEST SUC. CODE
5762								
5763	011266	000414				BR	8\$;ALL DONE
5764								
5765	011270	012704	000022		10\$:	MOV	#BDCHK,R4	;CHECKSUM ERROR
5766	011274	004737	012654			CALL	LOG	
5767	011300	000407				BR	8\$;EXIT
5768								
5769	011302	005302			7\$:	DEC	R2	;ANY PACKETS LEFT TO CHECK?
5770	011304	001405				BEQ	8\$;NO, ALL DONE
5771	011306	063700	003320			ADD	RCBCNT,R0	;YES, POINT TO NEXT PACKET
5772	011312	022121				CMP	(R1)+,(R1)+	;POINT TO NEXT EXPECTED COUNT
5773	011314	022323				CMP	(R3)+,(R3)+	;AND EXPECTED FLAG
5774	011316	000620				BR	1\$;TRY ANOTHER,THEY'RE SMALL
5775	011320	000207			8\$:	RETURN		;RETURN

```

5778 .SBTTL CHKEND / CHECK SUCCESS AND DETERMINE RETRY STATUS /
5779
5780 ;**
5781 ; CHKEND - IF RETRYING, DETERMINE IF DATA ERROR OR BAD FLAG BYTE ERROR RETRY.
5782 ;
5783 ; IF RETRYING BAD FLAG: RESET RETRY FLAG (SINCE OPERATION IS COMPLETE),
5784 ; AND CHECK SUCCESS CODE.
5785 ; IF RETRYING DATA ERROR; CHECK SUCCESS CODE AND IF 0, PRINT RECOVERED,
5786 ; SOFT ERROR, END RETRY STATUS. IF NOT 0 AND WAS STILL "DATA
5787 ; CHECK" ERROR - DETERMINE WHETHER TO CONTINUE ANOTHER RETRY OR
5788 ; LOG "UNRECOVERABLE" ERROR.
5789 ;
5790 ; IF NOT RETRYING DATA ERROR; CHECK IF 'DATA CHECK' ERROR SUCCESS CODE
5791 ; AND IF SO, START RETRY, ELSE EXIT.
5792 ; INPUTS: IMPLIED UNITS DATA BLOCK
5793 ; OUTPUTS: RETRY (SYSTAT BIT 1 AND 2), (BIT 10 OR 5) RESET IF RETRYING.
5794 ; - DATA COMPARE ERROR (BIT 6 OR 5) CLEARED.
5795 ; - REDUCED/NORMAL GAIN (BIT 7 OR 5) ADJUSTED
5796 ;--
5797
5798 011322 000240
5799 011324
(1) 011324 010046
(1)
(1)
5800 011326
(1) 011326 010446
(1)
(1)
5801 011330 032737 000006 003310
5802 011336 001406
5803 011340 032737 000004 003310
5804 011346 001454
5805 011350 042715 020000
5806 011354 004737 012340
5807 011360 032715 1C 0000
5808 011364 001402
5809 011366 000137 012044
5810 011372 105765 000077
5811 011376 001013
5812 011400 032715 000100
5813 011404 001002
5814 011406 000137 012044
5815 011412 012704 000014
5816 011416 004737 012654
5817 011422 000137 012044
5818 011426 032715 001000
5819 011432 001002
5820 011434 000137 012044
5821 011440 052715 002000
5822 011444 012765 000001 000002
5823 011452
(8) 011452 016546 000002
(7) 011456 012746 012224
(6) 011462 012746 000002
(3) 011466 010600

CHKEND:; NOP
PUSH R0 ;R0 --> END PAK
MOV R0, -(SP)

PUSH R4 MOV R4, -(SP)

1$: BIT #BIT1:BIT2, SYSTAT ;RETRYING?
DEQ NOREE ;NO-CHECK NORMALLY
BIT #BIT2, SYSTAT ;IS IT BAD FLAG TYPE?
BEQ CHKREE ;NO(DATA TYPE)
BIC #BIT13, R5 ;YES, SO IF END PACK THEN RETRY'S COMPLETE
NOREE: CALL CHKSUC ;CHECK SUCCESS CODE
BIT #BIT15, R5 ;ABORTED?
BEQ 3$ ;NO, CONTINUE
JMP CHKRET ;YES, EXIT
3$: TSTB SUCCS+1(R5) ;NO, HOW'D WE DO?
BNE CHKERR ;NOT SO GOOD.
BIT #BIT6, R5 ;OK, MOST FIND DATA PAK ERROR?
BNE 2$ ;YES
JMP CHKRET ;NO
2$: MOV #BDCOM, R4 ;YES, JUST BAD DATA-NO DATACHK ERR
CALL LOG ;BAD DATA IN PACKET
JMP CHKRET ;QUIT
CHKERR: BIT #BIT9, R5 ;BAD SUCCESS: TU DATA CHK ERROR?
BNE 1$ ;YES
JMP CHKRET ;NO, ALL DONE.
1$: BIS #BIT10, R5 ;YES-START RETRY
MOV #1, RETRY(R5) ;CALL IT 1ST
PRINTX #RTRYN, RETRY(R5) ;** PRINT **

MOV RETRY(R5)
MOV #RTRYN,
MOV #2, -(SP)
MOV SP, R0

```


(4)	011470	104415								
(4)	011472	062706	000506						TRAP	C\$PNTX
5824	011476	000562							ADD	#6, SP
5825	011500	004737	012340	CHKREE:	BR	CHKRET		; ALL DONE		
5826	011504	105765	000077		CALL	CHKSUC		; CHECK SUCCESS CODE		
5827	011510	001054			TSTB	SUCCS+1(R5)		; SUCCESSFUL YET?		
5828	011512				BNE	UNsuc		; NO, CHECK COUNT		
(8)	011512	016546	000002		PRINTX	#RECOV,RETRY(R5)				
(7)	011516	012746	012064						MOV	RETRY(R5
(6)	011522	012746	000002						MOV	#RECOV, -
(3)	011526	010600							MOV	#2, -(SP)
(4)	011530	104415							MOV	SP, R0
(4)	011532	062706	000006						TRAP	C\$PNTX
5829	011533	105715							ADD	#6, SP
5830	011540	100411			TSTB	(R5)		; DETERMINE THRESHOLD		
5831	011542				BMI	2\$; IT'S MODIFIED		
(7)	011542	012746	012144		PRINTX	#THRSLO		; NORMAL		
(6)	011546	012746	000001						MOV	#THRSLO,
(3)	011552	010600							MOV	#1, -(SP)
(4)	011554	104415							MOV	SP, R0
(4)	011556	062706	000004						TRAP	C\$PNTX
5832	011562	000410							ADD	#4, SP
5833	011564				BR	3\$				
(7)	011564	012746	012172	2\$:	PRINTX	#THRSHI		; ENHANCED		
(6)	011570	012746	000001						MOV	#THRSHI,
(3)	011574	010600							MOV	#1, -(SP)
(4)	011576	104415							MOV	SP, R0
(4)	011600	062706	000004						TRAP	C\$PNTX
5834	011604	032715	000400						ADD	#4, SP
5835	011610	001003		3\$:	BIT	#BIT8, #R5		; WRITE OR READ OPERATION?		
5836	011612	012704	000002		BNE	4\$; WRITE		
5837	011616	000402			MOV	#SFTRD, R4		; READ		
5838	011620	012704	000004		BR	5\$				
5839	011624	004737	012654	4\$:	MOV	#SFTWR, R4		; WRITE		
5840	011630	005065	000002	5\$:	CALL	LOG				
5841	011634	042715	002200		CLR	RETRY(R5)		; RESTORE TO NORMAL STATE		
5842	011640	000501			BIC	#BIT10!BIT7, #R5		; NO RETRY, NORM THRESHOLD		
5843					BR	CHKRET		; QUIT		
5844	011642	000240		UNsuc:	NOP			; RETRYING; SEE IF HARD YET		
5845	011644	032715	001000		BIT	#BIT9, #R5		; TU DATA CHECK ERROR?		
5846	011650	001015			BNE	2\$; YES		
5847	011652				PRINTB	#RETERR		; NO-"OTHER-ERROR" ERROR		
(7)	011652	012746	012266						MOV	#RETERR,
(6)	011656	012746	000001						MOV	#1, -(SP)
(3)	011662	010600							MOV	SP, R0
(4)	011664	104415							TRAP	C\$PNTB
(4)	011666	062706	000004						ADD	#4, SP
5848	011672	005065	000002		CLR	RETRY(R5)		; NO RETRIES		
5849	011676	042715	002200		BIC	#BIT10!BIT7, #R5		; NO RETRY, NORM THRESHOLD		
5850	011702	000460			BR	CHKRET		; EXIT		
5851	011704	023765	003332 000002	2\$:	CMP	MXRTRY, RETRY(R5)		; YES, DID WE GRADUATE TO HARD?		
5852	011712	001425			BEQ	HRD1		; YES		
5853	011714	005265	000002		INC	RETRY(R5)		; NO. JUST ANOTHER		
5854	011720				PRINTX	#RTRYN, RETRY(R5)		; PRINT OUT		
(8)	011720	016546	000002						MOV	RETRY(R5
(7)	011724	012746	012224						MOV	#RTRYN, -

5855	011744	032715	000200		BIT	#BIT7,SR5		; WAS NORMAL THRESHOLD?
5856	011750	001403			BEQ	1		; YES-REDUCE GAIN
5857	011752	042715	000200		BIC	#BIT7,SR5		; NO-NORM
5858	011756	000432			BR	CHKRET		
5859	011760	052715	000200	1:	BIS	#BIT7,SR5		; REDUCED
5860	011764	000427			BR	CHKRET		; DONE
5861	011766	000240		HRD1:	NOP			; HERE IS HARD ERROR!
5862	011770				PRINTX	#UNREC		
5863	012010	032715	000400		BIT	#BIT8,SR5		; RD OR WR?
5864	012014	001003			BNE	4		; WRITE
5865	012016	012704	000016		MOV	#HRDRD,R4		; READ
5866	012022	000402			BR	5		; LOG IT
5867	012024	012704	000020	4:	MOV	#HRDWR,R4		; WRITE
5868	012030	004737	012654	5:	CALL	LOG		; LOG IT
5869	012034	005065	000002		CLR	RETRY(R5)		; BACK TO NORMAL
5870	012040	042715	002200		BIC	#BIT10!BIT7,SR5		; NO RETRY, NOT REDUCED
5871								
5872	012044	042737	000006	003310	CHKRET:	BIC	#BIT1!BIT2,SYSTAT	; NO SYSTEM RETRY NEXT PASS
5873	012052	042715	000100		BIC	#BIT6,SR5		; NO MORE HOST DATA CHECK ERROR
5874	012056				PCP	R4		
5875	012060				POP	R0		MOV (SP)+,R4
5876	012062	000207			RETURN			MOV (SP)+,R0
5877								
5878								
5879	012064	040445	042522	047503	RECOV:	.ASCIZ	/#ARECOVERED FROM DATA CHECK ERROR RETRY #D1#N/	
5880						.EVEN		
5881	012144	040445	047040	051117	THRSLO:	.ASCIZ	/#A NORMAL THRESHOLD#N/	
5882						.EVEN		
5883	012182	040445	046440	042117	THRSHI:	.ASCIZ	/#A MODIFIED THRESHOLD #N/	
5884		012224				.EVEN		
5885	012224	040445	042522	051124	RTRYN:	.ASCIZ	/#ARETRY #D1#N/	
5886						.EVEN		
5887	012244	040445	047125	042522	UNREC:	.ASCIZ	/#AUNRECOVERABLE#N/	
5888						.EVEN		
5889	012266	040445	052117	042510	RETCRR:	.ASCIZ	/#AOTHER ERROR DURING RETRY ; EXIT RETRY#N/	
5890						.EVEN		

```

5893          ,SBTTL CHKSUC / INTERPRET SUCCESS CODE /
5894
5895          ***
5896          ; CHKSUC - COPY SUCCESS CODE (BYTE) TO SUCCS+1(R5). INTERPRET SUCCESS
5897          ; AND IF NOT 0, LOG APPROPRIATE ERROR.
5898          ; INPUTS: R0 POINTS TO END PACKET.
5899          ;          BR5 - UNIT STATUS WORD
5900          ;          CMD5NT(R5) - COMMAND BYTE
5901          ;
5902          ; OUTPUTS: R4 IS ERROR NUMBER IF ERROR.
5903          ;          SUCCS(R5) UPDATED.
5904          ;          BIT9 BR5 SET ON DATA CHECK SUCCESS CODE
5905          ;--
5906
5907          CHKSUC: NOP
5908          MOV      2(R0),SUCCS(R5) ;R0-->NEW PACKET
5909          CMPB    #ESOK,3(R0)      ;GET SUCCESS BYTE
5910          BEQ     12$              ;COMPLETE SUCCESS-EXIT
5911
5912          CMPB    #ESTRY,3(R0)     ;OK BUT RETRIES?
5913          BNE     20$              ;NO
5914          CMPB    CMD5NT(R5),#RSSRD ;A READ?
5915          BNE     22$              ;NO
5916
5917          BR      10$              ;NO RETRIES IN MAINTENANCE!
5918          22$:  CMPB    CMD5NT(R5),#RSSWR ;A WRITE?
5919          BNE     20$              ;NO
5920          BR      10$              ;LOG 1;
5921          20$:  CMPB    #ESNOMO,3(R0) ;NO MOTOR?
5922          BNE     1$              ;NO
5923          MOV     #NOMOT,R4        ;YES-
5924          BR      11$              ;LOG
5925
5926          1$:   CMPB    #ESCKS,3(R0) ;"DATA CHECK" ERROR?
5927          BNE     2$              ;NO
5928          BIS     #BIT9,BR5        ;SET DATA-CHK-ERROR FLAG
5929          BR      12$              ;DONT LOG
5930
5931          2$:   CMPB    CMD5NT(R5),#RSSSLF ;SELF TEST?
5932          BNE     3$              ;NOPE
5933          TSTB   3(R0)             ;YES, NEG. IF ERROR
5934          BPL     12$              ;OK
5935
5936          MOV     #SLFER,R4        ;YES-ERROR
5937          BR      11$              ;LOG IT
5938
5939          3$:   CMPB    #ESSK,3(R0)  ;SEEK ERROR?
5940          BNE     4$              ;NO
5941          MOV     #SKERR,R4        ;YES-
5942          BIS     #BIT14,BR5       ;SET 'DOBRK' FLAG *** REV E: *** MISSING "0"
5943          BR      11$              ;LOG
5944
5945          4$:   CMPB    #ESNCRT,3(R0) ;NO CART?
5946          BNE     5$              ;NO
5947          MOV     #NCART,R4        ;YES-
5948          BR      11$              ;LOG

```

5949								
5950	012534	122760	177720	000003	5#:	CMPB	#ESCMD,3(R0)	;NO UNDERSTAND HOST?
5951	012542	001003				BNE	6#	;NO
5952	012544	012704	000040			MOV	#CMNDR,R4	;YES-
5953	012550	000436				BR	11#	;LOG
5954								
5955	012552	122760	177770	000003	6#:	CMPB	#ESNONX,3(R0)	;NON EXISTENT UNIT?
5956	012560	001003				BNE	7#	;NO
5957	012562	012704	000036			MOV	#NUNIT,R4	;YES-
5958	012566	000427				BR	11#	;LOG
5959								
5960	012570	122760	177765	000003	7#:	CMPB	#ESWLOC,3(R0)	;WRITE LOCKED?
5961	012576	001003				BNE	8#	;NO
5962	012600	012704	000026			MOV	#WRLOCK,R4	;YES-
5963	012604	000420				BR	11#	;LOG
5964								
5965	012606	122760	177776	000003	8#:	CMPB	#ESPART,3(R0)	;PARTIAL OP?
5966	012614	001003				BNE	9#	;NO
5967	012616	012704	000034			MOV	#PARTL,R4	;YES-
5968	012622	000411				BR	11#	;LOG
5969								
5970	012624	122760	177711	000003	9#:	CMPB	#ESREC,3(R0)	;WRONG RECORD?
5971	012632	001003				BNE	10#	;NO
5972	012634	012704	000042			MOV	#RECERR,R4	;YES-
5973	012640	000402				BR	11#	;LOG
5974								
5975	012642	012704	000046		10#:	MOV	#SUCOTL,R4	;UNDEFINED
5976	012646	004737	012654		11#:	CALL	LOG	;LOG ERROR
5977	012652	000207			12#:	RETURN		;RETURN

```

5980 .SBTTL LOG / TO LOG ERROR IN CORRECT PLACE
5981
5982 ;**
5983 ; LOG - DETERMINE IF ERROR IS FATAL, NON-FATAL OR FATAL AFTER N TRIES
5984 ; BY INDEX (ERROR #) INTO DEVICE DATA BLOCK. ADD THE DRIVE # TO
5985 ; INDICATE UPPER OR LOWER BYTE AND INCREMENT THAT ERROR UNLESS
5986 ; THAT BYTE WOULD OVERFLOW. DETERMINE IF EVL FLAG SET, AND IF SO,
5987 ; CHECK THRESHOLD (EVLTHR) AND PRINT APPROPRIATE ERROR MESSAGE
5988 ; DESCRIPTION. ABORT THE UNIT IF INDICATED THROUGH DODROP CODE.
5989 ; INPUTS: R4 = ERROR CODE
5990 ; OUTPUTS: ABNDX(R5) = ERROR CODE.
5991 ; DLV(R5) = 0
5992 ; L#LUN = UNIT NUMBER
5993 ;--
5994
5995 012654 LOG:: PUSH R0
5996 (1) 012654 010046 MUV R0, -(SP)
5997 (1)
5998 (1)
5999
6000 012656 PUSH R1
6001 (1) 012656 010146 MOV R1, -(SP)
6002 (1)
6003 (1)
6004
6005 012660 PUSH R3
6006 (1) 012660 010346 MOV R3, -(SP)
6007 (1)
6008 (1)
6009
6010 012662 PUSH R4
6011 (1) 012662 010446 MOV R4, -(SP)
6012 (1)
6013
6014 012664 011537 002074 MOV @R5, L#LUN ;GET UNIT NUMBER
6015 012670 042737 177770 002074 BIC @177770, L#LUN ;MASK IT OFF
6016 012676 010465 000004 MOV R4, ABNDX(R5) ;SAVE INDEX IN CASE OF ABORT MESSAGE
6017 012702 012703 000120 MOV @LGOFST, R3 ;OFFSET TO LOW ORDER BYTE (DRIVE0)
6018 012706 060403 ADD R4, R3 ;FORM INDEX OF PARAM. TO UPDATE
6019 012710 060503 ADD R5, R3 ;FORM ABSOLUTE ADDR. THIS UNIT
6020 012712 004737 013660 CALL WHCHDR ;SEE WHICH DRIVE T' WAS
6021 012716 103001 BCC 2# ;WAS DRIVE 0
6022 012720 005203 INC R3 ;DRIVE 1; POINT TO UPPER BYTE
6023 012722 122713 000377 2#: CMPB @255., @R3 ;POTENTIAL OVERFLOW POSSIBLE?
6024 012726 001005 LOGO: BNE LOGOK ;NO
6025 012730 (4) 012730 104455 LOGO: ERRDF 0., OVRFLO, ERDES ;YES
6026 (5) 012732 000000 TRAP C#ERDF
6027 (5) 012734 013554 .WORD 0
6028 (5) 012736 013210 .WORD OVRFLO
6029 012740 000512 .WORD ERDES
6030 012742 105213 LOGOK: BR ABO ;ABORT UNIT
6031 012744 111304 MOV @R3, R4 ;INCREMENT THE ERROR
6032 012746 016503 000004 MOV ABNDX(R5), R3 ;TEMP'LY SAVE IT
6033 012752 012701 002230 MOV @RSNTAB, R1 ;GET INDEX AGAIN
6034 012756 066501 000004 ADD ABNDX(R5), R1 ;FORM ADRS OF MSG
6035 012762 042701 000001 BIC @BIT0, R1 ;LIKE THIS
6036 012766 032737 000004 016774 BIT @EVL, FLGLOC ;INSURE WORD BOUNDARY
;EVL SELECTED?

```

6020	012774	001414		BEQ	LOGOK2		;NO-CONT		
6021	012776	123704	002222	CMPB	EVLTHR,R4		;YES,OVER THRESHOLD?		
6022	013002	101011		BHI	LOGOK2		;NO		
6023	013004	010337	013016	MOV	R3,DFTL1+2		;YES,LOAD ERROR #		
6024	013010	011137	013020	MOV	BR1,DFTL1+4		;AND MESSAGE ADDR		
6025	013014			DFTL1:	ERRDF	0,DFTL1,ERRDES	;ERROR		
(4)	013014	104455						TRAP	C\$ERDF
(5)	013016	000000						.WORD	0
(5)	013020	013014						.WORD	DFTL1
(5)	013022	013210						.WORD	ERRDES
6026	013024	000460		BR	ABO		;DROP IT		
6027	013026	120327	000014	LOGOK2:	CMPB	R3,#BDCOM	;NEVER FATAL' TYPE?		
6028	013032	103011		BHIS	NTSFT		;NO		
6029	013034	010337	013046	MOV	R3,LOG1+2		;YES, ERROR CODE		
6030	013040	011137	013050	MOV	BR1,LOG1+4		;DESCRIPTION		
6031	013044			LOG1:	ERRSOFT	0,LOG1,ERRDES			
(4)	013044	104457						TRAP	C\$ERSOFT
(5)	013046	000000						.WORD	0
(5)	013050	013044						.WORD	LOG1
(5)	013052	013210						.WORD	ERRDES
6032	013054	000450		BR	LOGO		;EXIT		
6033									
6034	013056	120327	000026	NTSFT:	CMPB	R3,#WRLOCK	;ONE TRY?		
6035	013062	103411		BLO	MABEE		;NO, MAYBE A MULTIPLE		
6036	013064	010337	013076	MOV	R3,LOG2+2		;YES		
6037	013070	011137	013100	MOV	BR1,LOG2+4				
6038	013074			LOG2:	ERRHRD	0,LOG2,ERRDES	;PRINT HARD MESSAGE		
(4)	013074	104456						TRAP	C\$ERHRD
(5)	013076	000000						.WORD	0
(5)	013100	013074						.WORD	LOG2
(5)	013102	013210						.WORD	ERRDES
6039	013104	000430		BR	ABO		;DROP UNIT		
6040									
6041	013106	042704	177400	MABEE:	BIC	#177400,R4	;NEGATE SIGN EXTEND		
6042	013112	163704	003322	1#:	SUB	FTLNM,R4	;SEE IF MULTIPLE OF		
6043	013116	001413		BEQ	HRD		;FTLNM-YES!		
6044	013120	103401		BLO	SFT		;NO		
6045	013122	000773		BR	1#		;NOT THERE YET		
6046									
6047	013124	010337	013136	SFT:	MOV	R3,LOG3+2	;ERROR CODE		
6048	013130	011137	013140	LOG3:	MOV	BR1,LOG3+4	;DESCRIPTION		
6049	013134				ERRSOFT	0,LOG3,ERRDES			
(4)	013134	104457						TRAP	C\$ERSOFT
(5)	013136	000000						.WORD	0
(5)	013140	013134						.WORD	LOG3
(5)	013142	013210						.WORD	ERRDES
6050	013144	000414		HRD:	BR	LOGO	;EXIT		
6051	013146	010337	013160	MOV	R3,LOG3B+2		;HARD ERROR CODE		
6052	013152	011137	013162	MOV	BR1,LOG3B+4		;DESCRIPTION		
6053	013156			LOG3B:	ERRHRD	0,LOG3B,ERRDES			
(4)	013156	104456						TRAP	C\$ERHRD
(5)	013160	000000						.WORD	0
(5)	013162	013156						.WORD	LOG3B
(5)	013164	013210						.WORD	ERRDES
6054									
6055	013166	011500		ABO:	MOV	BR5,R0	;GET UNIT NUMBER		

6056	013170	042700	177770		BIC	#177770,R0					
6057	013174				DODU	R0					
(3)	013174	104451									
6058	013176			LOGO:	POP	R4				TRAP	C\$DODU
(1)	013176	012604									
(1)											
6059	013200				POP	R3					
(1)	013200	012603									
(1)											
6060	013202				POP	R1					
(1)	013202	012601									
(1)											
6061	013204				POP	R0					
(1)	013204	012600									
(1)											
6062	013206	000207			RETURN						

```

6065
6066
6067
6068
6069
6070 013210          BGNMSG  ERRDES          ;ERROR DESCRIPTION
      (3) 013210          ;ERRDES::
6071 013210 010046   PUSH      R0          MOV      R0,-(SP)
      (1) 013210
      (1)
6072 013212 010246   PUSH      R2          MOV      R2,-(SP)
      (1) 013212
      (1)
6073 013214 005002   CLR      R2          ;PRESET TO DATA TYPE
6074 013216 032715 000020  BIT      @BIT4,@R5   ;WHAT PACK TYPE?
6075 013222 001401   BEQ     2#          ;DATA
6076 013224 005202   INC     R2          ;COMMAND
6077 013226          2#: PRINTB @UNIT,<B,DR(R5)>,R2,<B,SYSTAT+1>
      (10) 013226 005046          CLR      -(SP)
      (10) 013230 153716 003311  BISB   SYSTAT+1
      (9) 013234 010246          MOV     R2,-(SP)
      (8) 013236 005046          CLR      -(SP)
      (8) 013240 156516 000060  BISB   DR(R5),@
      (7) 013244 012746 013402  MOV     @UNIT,-(
      (6) 013250 012746 000004  MOV     @4,-(SP)
      (3) 013254 010600          MOV     SP,R0
      (4) 013256 104414          TRAP   C#PNTB
      (4) 013260 062706 000012  ADD    @12,SP
6078 013264 016500 000064  MOV     REC(R5),R0   ;RECORD NUMBER
6079 013270 016502 000072  MOV     PATTEN(R5),R2 ;DATA EXPECTED
6080 013274          PRINTB @RECID,R0,<B,CMDSENT(R5)>,<B,R2>,<B,SUCCS+1(R5)>
      (11) 013274 005046          CLR      -(SP)
      (11) 013276 156516 000077  BISB   SUCCS+1(
      (10) 013302 005046          CLR      -(SP)
      (10) 013304 150216          BISB   R2,(SP)
      (9) 013306 005046          CLR      -(SP)
      (9) 013310 156516 000100  BISB   CMDSENT(R
      (8) 013314 010046          MOV     R0,-(SP)
      (7) 013316 012746 013462  MOV     @RECID,-
      (6) 013322 012746 000005  MOV     @5,-(SP)
      (3) 013326 010600          MOV     SP,R0
      (4) 013330 104414          TRAP   C#PNTB
      (4) 013332 062706 000014  ADD    @14,SP
6081 013336 005765 000074  TST     DLV(R5)      ;DLV ERROR?
6082 013342 001414   BEQ     3#          ;NO
6083 013344          PRINTB @RECID2,DLV(R5) ;YES-PRINT
      (8) 013344 016546 000074          MOV     DLV(R5),
      (7) 013350 012746 013636  MOV     @RECID2,
      (6) 013354 012746 000002  MOV     @2,-(SP)
      (3) 013360 010600          MOV     SP,R0
      (4) 013362 104414          TRAP   C#PNTB
      (4) 013364 062706 000006  ADD    @6,SP
6084 013370 005065 000074  CLR     DLV(R5)      ;RESET
6085 013374          3#: POP     R2          ;RESTORE

```


(1) 013374 012602
 (1)
 6086 013376
 (1) 013376 012600
 (1)
 6087 013400
 (3) 013400
 (3) 013400 104423
 6088 013402 040445 051104 053111
 6089
 6090 013462 040445 046102 041517
 6091 013554
 6092 013554 040503 023516 020124
 6093 013636
 6094 013636 040445 051040 042103
 6095

MOV (SP)+,R2
 POP R0
 MOV (SP)+,R0
 ENDMSG ;EXIT

L10003: TRAP C+MSG

UNIT:: .ASCIZ /#ADRI# #01#A PAK SENT #01#A FLAG RCVD #03#N/
 .EVEN
 RECI1:: .ASCIZ /#ABLOCK# #04#A COMMAND #02#A EXPCTD #03#A SUCCESS #03#N/
 .EVEN
 OVRFL0: .ASCIZ /CAN'T UPDATE ERROR OR STATISTIC:OVERFLOW PENDING/
 .EVEN
 RECI2: .ASCIZ /#A RCDB WAS #06#N/
 .EVEN

```

6098 .SBTTL WHCHDR / SEE WHICH DRIVE IS ACTIVE
6099
6100
6101 ;**
6102 ; INPUTS: DR(R5)
6103 ; OUTPUTS: CARRY*DRIVE (1 OR 0)
6104 ;--
6105
6106 013660 000241 WHCHDR:: CLC ;CLEAR CARRY
6107
6108 013662 105765 000060 TSTB DR(R5) ;DR 0?
6109 013666 001401 BEQ 2$ ;YES
6110 013670 000261 SEC ;NO
6111
6112 013672 000207 2$: RETURN ;RETURN

```

```

6115          .SBTTL  CHKSUM / FORM THE PACKET CHECKSUM
6116
6117          ;++
6118          ; THE CHECKSUM IS A 16 BIT CHECKSUM WITH END-AROUND CARRY.
6119          ;
6120          ; INPUTS:  R0 -> (POINTS TO) TOP OF PACKET
6121          ;           R1 = # OF BYTES
6122          ; OUTPUTS: R0 -> WHERE TO PUT CHECKSUM
6123          ;           R1 = CHECKSUM
6124          ;--
6125
6126
6127 013674      CHKSUM:: PUSH  R3
(1) 013674 010346      MOV   R3, -(SP)
(1)
(1)
6128 013676      PUSH  R2
(1) 013676 010246      MOV   R2, -(SP)
(1)
(1)
6129 013700 042737 000001 003310      BIC   #BIT0,SYSTAT ;"CHECKSUM IS ODD" BIT
6130 013706 032701 000001              BIT   #BIT0,R1      ;AN ODD # OF BYTES?
6131 013712 001403                      BEQ   1$           ;NO
6132 013714 052737 000001 003310      BIS   #BIT0,SYSTAT ;YES
6133
6134 013722 006001      1$:  ROR   R1           ;/2 FOR WORDS
6135
6136 013724 005003      2$:  CLR   R3           ;PREP CHECKSUM WORD
6137
6138 013726 062003      3$:  ADD   (R0)+,R3      ;FORM SUM
6139 013730 005503              ADC   R3           ;WITH CARRY
6140 013732 005301              DEC   R1           ;MORE WORDS?
6141 013734 001374              BNE   3$           ;YES
6142
6143 013736 032737 000001 003310      BIT   #BIT0,SYSTAT ;WAS IT ODD
6144 013744 001405              BEQ   4$           ;NO
6145 013746 112002              MOVB  (R0)+,R2     ;YES GET NEXT BYTE
6146 013750 042702 177400      BIC   #177400,R2   ;UN-SIGN EXTEND
6147 013754 060203              ADD   R2,R3       ;ADD IT IN
6148 013756 005503              ADC   R3           ;AND CARRY JUST IN CASE
6149
6150 013760 010301      4$:  MOV   R3,R1       ;RETURN IT IN CORRECT PLACE
6151 013762              POP   R2           ;RESTORE
(1) 013762 012602      MOV   (SP)+,R2
(1)
6152 013764              POP   R3
(1) 013764 012603      MOV   (SP)+,R3
(1)
6153 013766 000207      RETURN          ;RETURN
    
```

```

6156 .SBTTL CKCKSM / MODULE TO CHECK THE CHKSUMS
6157
6158
6159 ; MAKE SURE THE CHECKSUM RECEIVED = THE CHECKSUM CALCULATED.
6160 ; INPUTS: R4 = THE PACKET BYTE COUNT
6161 ; RO -> THE PACKET TOP
6162 ; OUTPUTS: CARRY SET IF CHECKSUM CALC'D DOES NOT EQUAL CHECKSUM SENT
6163 ; RO -> THE PACKET TOP
6164 ;--
6165
6166
6167 013770 CKCKSM:: PUSH R1
(1) 013770 010146 MOV R1, -(SP)
(1)
(1)
6168 013772 PUSH RO ;SAVE
(1) 013772 010046 MOV RO, -(SP)
(1)
(1)
6169 013774 010401 MOV R4, R1 ;COPY BYTE COUNT TO CORRECT
6170 013776 004737 013674 CALL CHKSUM ;REGISTER FOR CHKSUM AND
;FORM CHECKSUM
6171
6172 ;HERE RO --> XMITTED CHKSUM, R1=CHKSUM CALC'D
6173
6174
6175 014002 122001 CMPB (RO)+, R1 ;LOWER ORDER CHECK
6176 014004 001005 BNE 2$ ;WRONG
6177
6178 014006 000301 SWAB R1 ;OK-PREP FOR
6179
6180 014010 122001 CMPB (RO)+, R1 ;HIGH ORDER CHECK
6181 014012 001002 BNE 2$ ;WRONG
6182 014014 000241 CLC ;OK-CLEAR SAILING
6183
6184 014016 000401 BR 3$ ;EXIT
6185
6186 014020 000261 2$: SEC ;LET ERROR BE KNOWN
6187
6188
6189 014022 3$: POP RO
(1) 014022 012600 MOV (SP)+, RO
(1)
6190 014024 POP R1
(1) 014024 012601 MOV (SP)+, R1
(1)
6191 014026 000207 RETURN ;RETURN
    
```

```

6194 .SBTTL DOBRK / MODULE TO INIT TUS8 AND TEST INTERRUPTS
6195
6196 ;++
6197 ; DOBRK - SEND RADIAL SERIAL "BREAK" TO DEVICE:
6198 ; - SET "BREAK" ON INTERFACE.
6199 ; - SEND 8. NULLS
6200 ; - CLEAR "BREAK" ON INTERFACE
6201 ; - SET VECTORS FOR RCV AND XMIT
6202 ; - SEND 2 BYTES OF "INIT"
6203 ; - RECEIVE "CONTINUE"
6204 ; - IF RECEIVE GARBAGE OR TIMEOUT - ERROR
6205 ; - CLEAR INTERRUPTS AND VECTORS
6206 ; INPUTS: BR5 BIT14 WAS SET - (SEND BREAK)
6207 ; OUTPUTS: BR5 BIT14 CLEAR IF SUCCESSFUL INIT.
6208 ;          SYSTAT+1 = RECEIVED BYTE
6209 ;          ERRORS R4 = ERROR CODE:
6210 ;          - SEND NOT READY TIMEOUT (TOSNDB)
6211 ;          - NO RESPONSE
6212 ;          - DLV ERROR
6213 ;          - CAN'T INIT
6214 ;--
6215
6216 DOBRK:: CLRB INITWD+1 ;CLEAR BYTE RECEIVE ADDR
6217 CLRB BRKTO ;CLEAR TIME OUT CONSTANT
6218 BIS #BIT0, @XMSR(R5) ;SET 'BREAK'
6219 MOV #RSSNIT, CMDSNT(R5) ;SAY WE SENT 'INIT'
6220 BIS #BIT4, BR5 ;PAK SENT TYPE =COMMAND, SORT OF
6221 MOV #8, R4 ;BREAK-IT'S-BACK COUNT=8
6222 1$: BREAK ;SUPERVISOR TAKE FIVE
6223 ; TRAP C$BRK
6224 ; FOR +C CHECK, ETC.
6225 TSTB @XMSR(R5) ;READY?
6226 BMI 4$ ;YES
6227 DEC BRKTO ;NO, TIME OUT?
6228 BNE 1$ ;NO
6229 MOV #TOSNDB, R4 ;YES, SET ERROR CODE
6230 CALL LOG ;LOG IT
6231 BR 3$ ;EXIT
6232 4$: MOVB BRKWD, @XMDB(R5) ;SEND NULL
6233 CLR BRKTO ;RESET TIME OUT
6234 DEC R4 ;MORE NULLS TO SEND?
6235 BNE 1$ ;YES
6236 DEC @XMSR(R5) ;NO, CLEAR 'BREAK'
6237 MOV @RCDB(R5), R0 ;HEAVE 'GARBAGE' 1ST BYTE
6238 SETPRI #PRI00 ;SET TO INTERRUPT FO SURE
6239 ; MOV #PRI00, R
6240 ; TRAP C$SPRI
6241 ; MOV #PRI07, -
6242 ; MOV #RCVINT,
6243 ; MOV TUVECT(R
6244 ; MOV #3, -(SP)
6245 ; TRAP C$SVEC
6246 ; ADD #10, SP
6247 SETVEC TUVECT(R5), #RCVINT, #PRI07 ;SET VECTO INFO
6248 ; MOV #PRI07, -
6249 ; MOV #RCVINT,
6250 ; MOV TUVECT(R
6251 ; MOV #3, -(SP)
6252 ; TRAP C$SVEC
6253 ; ADD #10, SP
6254 ADD #4, TUVECT(R5) ;AND INC TO SND VECTOR
6255 SETVEC TUVECT(R5), #SNDINT, #PRI07 ;AND SET IT

```

Address	Label	Op1	Op2	Op3	Op4	Op5	Op6	Op7	Op8
(7)	014204	012746	000340						
(6)	014210	012746	014454						MOV #PRI07,-
(5)	014214	016546	000204						MOV #SNDINT,
(4)	014220	012746	000003						MOV TUVECT(R
(3)	014224	104437							MOV #3,-(SP)
(2)	014226	052706	000010						TRAP C\$SVEC
6241	014232	162765	000004	000204		SUB	#4,TUVECT(R5)		ADD #10,SP
6242	014240	005037	014564			CLR	BRKTO		
6243	014244	012704	014562			MOV	#INITWD,R4		
6244	014250	010437	014566			MOV	R4,BRKPTR		
6245	014254	052775	000100	000026		BIS	#BIT6,#XMSR(R5)		
6246	014262	004737	014524			CALL	WAIT		
6247	014266	005715				TST	#R5		
6248	014270	100446				BMI	3\$		
6249									
6250	014272	005037	014564			CLR	BRKTO		
6251	014276	012704	014562			MOV	#INITWD,R4		
6252	014302	010437	014566			MOV	R4,BRKPTR		
6253	014306	052775	000100	000026		BIS	#BIT6,#XMSR(R5)		
6254	014314	004737	014524			CALL	WAIT		
6255	014320	005715				TST	#R5		
6256	014322	100431				BMI	3\$		
6257									
6258	014324	012704	014563			MOV	#INITWD+1,R4		
6259	014330	010437	014566			MOV	R4,BRKPTR		
6260	014334	052775	000100	000022		BIS	#BIT6,#RCSR(R5)		
6261	014342	004737	014524			CALL	WAIT		
6262	014346	005715				TST	#R5		
6263	014350	100416				BMI	3\$		
6264									
6265	014352	123727	014563	000020		CMPB	INITWD+1,#RSCONT		
6266	014360	001003				BNE	2\$		
6267									
6268	014362	042715	040000			BIC	#BIT14,#R5		
6269	014366	000407				BR	3\$		
6270									
6271	014370	113737	014563	003311		MOVB	INITWD+1,SYSTAT+1		
6272	014376	012704	000032			MOV	#CNINIT,R4		
6273	014402	004737	012654			CALL	LOG		
6274									
6275									
6276	014406	042775	000100	000026		BIC	#BIT6,#XMSR(R5)		
6277	014414	042775	000100	000022		BIC	#BIT6,#RCSR(R5)		
6278	014422					CLRVEC	TUVECT(R5)		
(3)	014422	016500	000204						
(3)	014426	104436							
6279	014430	062765	000004	000204		ADD	#4,TUVECT(R5)		
6280	014436					CLRVEC	TUVECT(R5)		
(3)	014436	016500	000204						
(3)	014442	104436							
6281	014444	162765	000004	000204		SUB	#4,TUVECT(R5)		
6282	014452	000207				RETURN			

```

        .SEYTL  INTERRUPT SERVICE ROUTINES AND TIMER
6285
6286
6287 014454      BGNSRV  SNDINT          ;"SEND" INTERRUPT SERVICE;
      (3) 014454                                SNDINT::
6288
6289 014454 042775 000100 000026      SNDHND: BIC    #BIT6,@XMSR(R5) ;DISABLE INTERRUPT
6290 014462 112475 000030              MOVB   (R4)+,@XMDB(R5);OUTPUT BYTE
6291 014466      ENDSRV
      (3) 014466                                L10004:
      (2) 014466 000002                                RTI
6292
6293
6294
6295 014470      BGNSRV  RCVINT          ;"RCV" INTERRUPT SERVICE;
      (3) 014470                                RCVINT::
6296
6297 014470 042775 000100 000022      RCVHND: BIC    #BIT6,@RCSR(R5) ;DISABLE INTS
6298 014476 017565 000024 000074      MOV    @RCD8(R5),DLV(R5) ;SAVE WORD
6299 014504 116524 000074              MOVB   DLV(R5),(R4)+ ;BYTE TO BUFFER
6300 014510 005765 000074              TST   DLV(R5) ;ERROR?
6301 014514 100402              BNE   10# ;YES
6302 014516 005065 000074              CLR   DLV(R5) ;NO CLEAR ERROR
6303 014522      10#:
6304 014522      ENDSRV
      (3) 014522                                L10005:
      (2) 014522 000002                                RTI
6305
6306
6307
6308 014524 000240      WAIT:  NOP          ;WAIT LOOP FOR
6309
6310 014526 020437 014566              CMP    R4,BRKPTR ;INTERRUPT SERVICING
6311 014532 001011              BNE   1# ;IF=,THEN NO INTERRUPT
6312 014534              BREAK ;GOT ONE!
      (3) 014534 104422              ;SUPERVISOR BREAK
6313 014536              BREAK ;KILL SOME TIME
      (3) 014536 104422              TRAP  C#BRK
6314 014540 005337 014564              DEC   BRKTO ;TIME OUT?
6315 014544 001367              BNE   WAIT ;NO...CONT.
6316 014546 012704 000050              MOV   #TOHCVB,R4 ;YES LOAD ERROR #
6317 014552 004737 012654              CL   LOG ;LOG IT
6318 014556 000207      1#:  RETURN ;RETURN
6319
6320 014560 000000      BRKWD: .WORD 0 ;NULL
6321 014562 004          INITWD: .BYTE RSINIT ;INIT COMMAND
6322 014563 000          ;.BYTE 0 ;RSCONT IS EXPECTED HERE
6323 014564 000000      BRKTO: .WORD 0 ;TIME OUT
6324 014566 000000      BRKPTR: .WORD 0 ;POINTER TO INITWD
    
```

```

6327          .SBTTL COMPAR/ DATA COMPARISON MODULE
6328
6329          ;;;
6330          ; COMPAR - IF "COMPARE DATA" SELECTED, COMPARE EACH DATA BYTE OF PACKET
6331          ; TO PATTEN(R5). SAVE NUMBER OF BYTES NOT CORRECT. IF NOT
6332          ; 0, PRINT SOFT ERROR AND TOTAL # WRONG BYTES. SET "BAD_DATA_
6333          ; IN_PACKET" BIT (BIT6 BR5) FOR HIGHER LEVEL MODULES.
6334          ; INPUTS:  - (CMPDAT) FLAG TO NOT COMPARE (=1)
6335          ;           - PKPTR(R5) POINTS TO DATA PACK.
6336          ; OUTPUTS: BIT6 BR5 (BAD DATA FLAG) ADJUSTED.
6337          ;           L$LUN - UNIT NUMBER
6338          ;           PRNSIZ - SIZE OF PACKET
6339          ;--
6340
6341          COMPAR:: PUSH    R0              ;COMPARE DATA IS DATA PACKET
6342          (1) 014570 010046             MOV     R0,-(SP)
6343          (1)
6344          (1)
6345          (1)
6346          014572 010446             PUSH   R4              ;TO PATTERN WRITTEN
6347          (1) 014572 010446             MOV     R4,-(SP)
6348          (1)
6349          (1)
6350          014574 010146             PUSH   R1              ;USING BYTE COUNT IN PACKET
6351          (1) 014574 010146             MOV     R1,-(SP)
6352          (1)
6353          014576 005037 014746         CLR     BDBYTS         ;CLEAR TOTAL WRONG
6354          014602 016504 000104         MOV     PKPTR(R5),R4  ;GET TOP OF PACKET
6355          014606 005737 002216         TST    CMPDAT         ;COMPARE SELECTED?
6356          014612 001451              BEQ     4#             ;NO-EXIT
6357          014614 005204              INC     R4             ;YES, LOCATE COUNT
6358          014616 111401              MOV     BR4,R1        ;GET IT
6359          014620 042701 177400         BIC     @177400,R1    ;SIGN-UNEXTEND
6360          014624 005204              ;MUST TEST BYTE-WISE...
6361          014626 126524 000072         1#; INC     R4         ;-->FIRST DATA BYTE
6362          014632 001402              CMPB   PATTEN(R5),(R4); ;DATA-WHAT WAS EXPECTED?
6363          014634 005237 014746         BEQ     2#             ;YES
6364          014640 005301              INC     BDBYTS        ;NO, INCREMENT TOTAL WRONG
6365          014642 001371              2#; DEC     R1        ;MORE LEFT?
6366          014644 005737 014746         SNE    1#             ;YES
6367          014650 001432              TST    BDBYTS        ;ANY WRONG?
6368          014652 011537 002074         BEQ     4#             ;NO
6369          014656 042737 177770 002074  MOV     BR5,L$LUN     ;GET UNIT NUMBER
6370          014664 104457              BIC     @177770,L$LUN ;MASK IT OFF
6371          (4) 014664 104457              ERRSOFT 0,MSBDA,ERRDES ;YES-PRINT 'BAD DATA IN PACKET' ERROR
6372          (5) 014666 000000              TRAP    C$ERSOFT
6373          (5) 014670 002342              .WORD  0
6374          (5) 014672 013210              .WORD  MSBDA
6375          014674              .WORD  ERRDES
6376          (8) 014674 013746 014746         PRINTB @DESC,BDBYTS
6377          (7) 014700 012746 014750         MOV     BDBYTS, .
6378          (6) 014704 012746 000002         MOV     @DESC,-(
6379          (3) 014710 010600              MOV     @2,-(SP)
6380          (4) 014712 104414              MOV     SP,R0
6381          (4) 014714 062706 000006         TRAP    C$PNTB
6382          ADD     @6,SP

```


D7

GLOBAL AREAS MACY11 30(1046) 25-JAN-84 08:33 PAGE 44-1
CZTUUF.P11 25-JAN-84 08:09 COMPAR/DATA COMPARISON MODULE

SEQ 0081

6364	014720	052715	000100		BIS	#BIT6,SR5	;LET 'EM KNOW UPSTAIRS-BAD DATA FLAG
6365	014724	012737	000204	003340	MOV	#132,PRNSIZ	;SIZE IS ONE DATA PACK
6366	014732	004737	015004		CALL	PRNPAK	;AND PRINT THE PACKET
6367	014736				POP	R1	;RESTORE
(1)	014736	012601					MOV (SP)+,R1
(1)							
6368	014740				POP	R4	
(1)	014740	012604					MOV (SP)+,R4
(1)							
6369	014742				POP	R0	
(1)	014742	012600					MOV (SP)+,R0
(1)							
6370							
6371	014744	000207			RETURN		
6372							
6373	014746	000000			BDBYTS:	.WORD	
6374	014750	040445	047524	040524	DESC:	.ASCIZ	/*ATOTAL BAD BYTES= #D3*A.#N/
6375						.EVEN	

```

6378 .SBTTL PRNPAK/MODULE TO PRINT DATA PACKET
6379
6380 ;**
6381 ; PRNPAK - IF PRINT_DATA_PACK_ON_ERROR SELECTED: PRINT EACH BYTE OF PACKET
6382 ; TO BY PKPTR(R5).
6383 ; INPUTS: PRNSIZ - # OF BYTES IN PACKET.
6384 ; OUTPUTS: NONE
6385 ;--
6386
6387 015004 000240 PRNPAK:: NOP ;PRINTS 1 PACKET
6388 ;PKPTR(R5)->TOP OF PACKET
6389 ;PRNSIZ (PASSED)=BYTE COUNT
6390 015006 PUSH R0
6391 (1) 015006 010046 MOV R0,-(SP)
6392 (1)
6393 (1)
6394 015010 PUSH R4
6395 (1) 015010 010446 MOV R4,-(SP)
6396 (1)
6397 (1)
6398 015012 105737 002214 TSTB PRBUF ;PRINT PACKET SELECTED?
6399 015016 001451 BEQ 4$ ;NO
6400 015020 016504 000104 MOV PKPTR(R5),R4 ;YES-GET TOP OF PACK
6401 015024 012737 000020 015150 1$: MOV #16,,LNCNT ;16 BYTES PER LINE
6402 015032 112437 015152 2$: MOVB (R4)+,PRDAT ;AVOID SIGN EXTEND
6403 015036 PRINTF @PRFORM,<B,PRDAT> ;PRINT BYTE
6404 (8) 015036 005046 CLR -(SP)
6405 (8) 015040 153716 015152 BISB PRDAT,(S
6406 (7) 015044 012746 015154 MOV @PRFORM,
6407 (6) 015050 012746 000002 MOV #2,-(SP)
6408 (3) 015054 010600 MOV SP,R0
6409 (4) 015056 104417 TRAP C$PNTF
6410 (4) 015060 062706 000006 ADD #6,SP
6411 6398 015064 005337 003310 DEC PRNSIZ ;ONE LESS
6412 6399 015070 001414 BEQ 3$ ;NO MORE
6413 6400 015072 005337 015150 DEC LNCNT ;NEW LINE?
6414 6401 015076 001355 BNE 2$ ;NOT YET
6415 6402 015100 PRINTF @CARLF ;YES
6416 (7) 015100 012746 015164 MOV @CARLF,-
6417 (6) 015104 012746 000001 MOV #1,-(SP)
6418 (3) 015110 010600 MOV SP,R0
6419 (4) 015112 104417 TRAP C$PNTF
6420 (4) 015114 062706 000004 ADD #4,SP
6421 6403 015120 000741 BR 1$ ;NEXT LINE
6422 6404 015122 3$: PRINTF @CARLF ;FINISH UP
6423 (7) 015122 012746 015164 MOV @CARLF,-
6424 (6) 015126 012746 000001 MOV #1,-(SP)
6425 (3) 015132 010600 MOV SP,R0
6426 (4) 015134 104417 TRAP C$PNTF
6427 (4) 015136 062706 000004 ADD #4,SP
6428 6405 015142 4$: POP R4
6429 (1) 015142 012604 MOV (SP)+,R4
6430 (1)
6431 6406 015144 POP R0
6432 (1) 015144 012600 MOV (SP)+,R0
6433 (1)

```

F7

6407	015146	000207			RETURN					;RETURN
6408										
6409	015150	000000			LNCNT:	.WORD				
6410	015152	000000			PRDAT:	.WORD				
6411	015154	047445	022463	020101	PRFORM:	.ASCIZ	/#03#A /			
6412		015164				.EVEN				
6413	015164	047045	000		CARLF:	.ASCIZ	/#N/			
6414		015170				.EVEN				
6415										
6416	015170									
6417						ENDMOD				

```

6430 .TITLE MISCELLANEOUS SECTIONS
6431 .SBTTL REPORT CODING SECTION
6459
6460 015170 BGNMOD
6461
6462
6463 ;++
6464 ; THE REPORT CODING SECTION CONTAINS THE
6465 ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
6466 ;--
6467 015170 BGNRPT
(3) 015170 L$RPT::
6468 015170 010046 PUSH R0 MOV R0, -(SP)
(1) 015170
(1)
6469 015172 010146 PUSH R1 MOV R1, -(SP)
(1) 015172
(1)
6470 015174 010246 PUSH R2 MOV R2, -(SP)
(1) 015174
(1)
6471 015176 010346 PUSH R3 MOV R3, -(SP)
(1) 015176
(1)
6472 015200 010446 PUSH R4 MOV R4, -(SP)
(1) 015200
(1)
6473 015202 010546 PUSH R5 MOV R5, -(SP)
(1) 015202
(1)
6474
6475 015204 BREAK
(3) 015204 104422
6476 015206 012737 003352 015616 MOV $BLKTB, RPTR ;GET 1ST DEVICE BLOCK TRAP C$BRK
6477 015214 PRINTS $STATHD ;HEADER
(7) 015214 012746 015620 MOV $STATHD,
(6) 015220 012746 000001 MOV $1, -(SP)
(3) 015224 010600 MOV SP, R0
(4) 015226 104416 TRAP C$PNTS
(4) 015230 062706 000004 ADD $4, SP
6478 015234 BREAK ;TC CHECK
(3) 015234 104422 TRAP C$BRK
6479 015236 PRINTS $STHD2 ;2ND HEADER
(7) 015236 012746 016074 MOV $STHD2,
(6) 015242 012746 000001 MOV $1, -(SP)
(3) 015246 010600 MOV SP, R0
(4) 015250 104416 TRAP C$PNTS
(4) 015252 062706 000004 ADD $4, SP
6480 015256 1$; BREAK ;TC CHECK
(3) 015256 104422 TRAP C$BRK

```

6481	015260	017705	000332		MOV	BRPTR,R5		;GET DEVICE BLOCK		
6482	015264	032715	004000		BIT	#BIT11,BR5		;UNIT NOT TESTED?		
6483	015270	001131			BNE	2#		;TRUE, DON'T PRINT STATISTICS		
6484								;OK TO PRINT		
6485	015272	011537	015614		MOV	BR5,RLUN		;SAVE STATUS WORD		
6486	015276	042737	177770	015614	BIC	#177770,RLUN		;MASK UNIT NUM.		
6487	015304	116501	000122		MOV	SCFTR(R5),R1		;SOFTREAD		
6488	015310	042701	177400		BIC	#177400,R1		;SIGN-UNEXTEND		
6489	015314	116502	000124		MOV	SOFTW(R5),R2		;SOFT WRITE		
6490	015320	042702	177400		BIC	#177400,R2				
6491	015324	116503	000136		MOV	HARDR(R5),R3		;HARD READ		
6492	015330	042703	177400		BIC	#177400,R3				
6493	015334	116504	000140		MOV	HARDW(R5),R4		;HARD WRITE		
6494	015340	042704	177400		BIC	#177400,R4				
6495	015344				PRINTS	#FM0,RLUN		;SUMMARY/UNIT #		
(8)	015344	013746	015614						MOV	RLUN,-(SP)
(7)	015350	012746	015732						MOV	#FM0,-(SP)
(6)	015354	012746	000002						MOV	#2,-(SP)
(3)	015360	010600							MOV	SP,R0
(4)	015362	104416							TRAP	C#PNTS
(4)	015364	062706	000006						ADD	#6,SP
6496	015370				PRINTS	#FM,#0,WRTNO(R5),RDNO(R5),<B,BDATA(R5)>,R1,R2,R3,R4				
(15)	015370	010446							MOV	R4,-(SP)
(14)	015372	010346							MOV	R3,-(SP)
(13)	015374	010246							MOV	R2,-(SP)
(12)	015376	010146							MOV	R1,-(SP)
(11)	015400	005046							CLR	-(SP)
(11)	015402	156516	000134						BISB	BDATA(R5
(10)	015406	016546	000114						MOV	RDNO(R5)
(9)	015412	016546	000110						MOV	WRTNO(R5)
(8)	015416	012746	000000						MOV	#0,-(SP)
(7)	015422	012746	015750						MOV	#FM,-(SP)
(6)	015426	012746	000011						MOV	#11,-(SP)
(3)	015432	010600							MOV	SP,R0
(4)	015434	104416							TRAP	C#PNTS
(4)	015436	062706	000024						ADD	#24,SP
6497	015442	116501	000123		MOV	SOFT+1(R5),R1		;SAME		
6498	015446	042701	177400		BIC	#177400,R1		;AS		
6499	015452	116502	000125		MOV	SOFT+1(R5),R2		;ABOVE		
6500	015456	042702	177400		BIC	#177400,R2		;THIS		
6501	015462	116503	000137		MOV	HARD+1(R5),R3		;TIME		
6502	015466	042703	177400		BIC	#177400,R3		;FOR		
6503	015472	116504	000141		MOV	HARD+1(R5),R4		;DRIVE		
6504	015476	042704	177400		BIC	#177400,R4		;ONE		
6505										
6506	015502				PRINTS	#FM,#1,WRTN1(R5),RDN1(R5),<B,BDATA+1(R5)>,R1,R2,R3,R4				
(15)	015502	010446							MOV	R4,-(SP)
(14)	015504	010346							MOV	R3,-(SP)
(13)	015506	010246							MOV	R2,-(SP)
(12)	015510	010146							MOV	R1,-(SP)
(11)	015512	005046							CLR	-(SP)
(11)	015514	156516	000135						BISB	BDATA+1(
(10)	015520	016546	000116						MOV	RDN1(R5)
(9)	015524	016546	000112						MOV	WRTN1(R5)
(8)	015530	012746	000001						MOV	#1,-(SP)
(7)	015534	012746	015750						MOV	#FM,-(SP)

```

(6) 015540 012746 000011
(3) 015544 010600
(4) 015546 104416
(4) 015550 062706 000024
6507 015554 023727 015616 003370 2*: CMP RPTR, #LSTDEV ;ALL UNITS DONE?
6508 015562 103005 3* BHS ;YES
6509 015564 062737 000002 015616 ADD #2,RPTR ;NO-DO
6510
6511 015572 000137 015256 JMP 1* ;MORE UNITS
6512
6513 015576 3*: POP R5 MOV (SP)+,R5
(1) 015576 012605
(1)
6514 015600 POP R4 MOV (SP)+,R4
(1) 015600 012604
(1)
6515 015602 POP R3 MOV (SP)+,R3
(1) 015602 012603
(1)
6516 015604 POP R2 MOV (SP)+,R2
(1) 015604 012602
(1)
6517 015606 POP R1 MOV (SP)+,R1
(1) 015606 012601
(1)
6518 015610 POP R0 MOV (SP)+,R0
(1) 015610 012600
(1)
6519 015612 ENDRPT
(3) 015612
(3) 015612 104425 L10006: TRAP C$RPT
6520 015614 000000
6521 015616 000000
6522
6523 0. 3620 047045 040445 020040
6524 0.5666 041504 045510 051057
6525 015732 .ASCII /#N#A DR BLKS WR BLKS RD BDPAK /
6526 015732 .ASCIZ @DCHK/RD DCHK/WR DCHK/RD DCHK/WR#N@ /
6527 015750 .EVEN
6528
6529 015750 .ASCII /#A #D1#A #D5#A. #D5#A. #D3#A. /
6530 016024 .ASCIZ /#D3#A. #D3#A. #D3#A. #D3#A.#N/ /
6531 016074 .EVEN
6532 016074 040445 020040 020040
6533 016141 122 041505 053117
6534 016204 .ASCII /#A
6535 016204 .ASCIZ /RECOV RECOV UNRECOV UNRECOV#N/ /
ENDMOD

```

```

6538 .SBTTL INITIALIZE SECTION
6539
6540
6541 ;**
6542 ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
6543 ; AT THE BEGINNING OF EACH PASS.
6544 ;--
6545 016204 BGNINIT
6546 (3) 016204 L$INIT::
6547 016204 000240 INIT: NOP ;
6551 016206 105037 016770 CLRB STRT ;FOR STATS CLEAR
6552 016212 005037 003344 CLR TEST9 ;***** CLR TST 9 FLAG
6553 016216 READEF #EF.START ;START COMMAND?
6554 (3) 016216 012700 000040 MOV #EF.START
6555 (3) 016222 104447 TRAP C$REFG
6556 016224 BNCOMPLETE INIT2 ;NO
6557 (2) 016224 103003 BCC INIT2
6558 016226 005237 016770 INC STRT ;YES, SET START FLAG
6559 (3) 016232 104433 BRESET ;BUSS RESET, EH?
6560 016234 012737 003352 003314 INIT2: MOV #BLKTBL,DEVPTR ;SET ALL UNITS ABORTED:
6561 016242 005004 CLR R4 ;UNIT NUMBER
6562 016244 017705 165044 1$: MOV #DEVPTR,R5 ;GET POINTER
6563 016250 010415 MOV R4,#R5 ;INSERT UNIT #
6564 016252 052715 100000 BIS #BIT15,#R5 ;SET ABORTED
6565 016256 052715 004000 BIS #BIT11,#R5 ;SET UNIT NOT TESTED
6566 016262 006304 ASL R4 ;*2 FOR LOOK-UP
6567 016264 016465 027724 000102 MOV BUFTBL(R4),RCVBUF(R5) ;SETUP POINTER TO UNIT'S BUFFER
6568 016272 006204 ASR R4 ;CORRECT BACK TO UNIT #
6569 016274 023727 003314 003370 CMP DEVPTR,#LSTDEV ;LAST DEVICE DONE?
6570 016302 103005 BHIS CHECK ;YES
6571 016304 062737 000002 003314 ADD #2,DEVPTR ;NO-GET
6572 016312 005204 INC R4 ;NEXT DEVICE AND
6573 016314 000753 BR 1$ ;SERVICE
6574 016316 022737 000010 002012 CHECK: CMP #8,,L$UNIT ;MAKE SURE NOT
6575 016324 103005 BHIS GETHRD ;TOO MANY UNITS
6576 (4) 016326 104454 ERRSF 101,,TOMANY ;TOMANY-REQUEST +C
6577 (5) 016330 000145 TRAP C$ERSF
6578 (5) 016332 016706 .WORD 101
6579 (5) 016334 000000 .WORD TOMANY
6580 016336 DOCLN ;EXIT
6581 (3) 016336 104444 TRAP C$DCLN
6582 016340 012737 003352 003314 GETHRD: MOV #BLKTBL,DEVPTR ;INIT TABLE POINTER
6583 016346 005004 CLR R4 ;CLEAR DEVICE COUNTER
6584 016350 017705 164740 1$: MOV #DEVPTR,R5 ;GET STATUS WORD
6585 016354 010437 002074 MOV R4,,LUN ;UNIT NUM. IN CASE ERROR
6586 016360 GPHARD R4,R2 ;GET HARD INFO
6587 (3) 016360 010400 MOV R4,R0
6588 (3) 016362 104442 TRAP C$GPHRD
6589 (3) 016364 010002 MOV R0,R2
6590 016366 BNCOMPLETE 3$
6591 (2) 016366 103111 BCC 3$

```

6583	016370	042715	004000		BIC	#BIT11,R5	;UNIT IS TESTED!		
6584	016374	012203			MOV	(R2)+,R3	;R3=CSR		
6585	016376	012265	000204		MOV	(R2)+,TUVECT(R5)	;GET VECTOR ADDRESS		
6586	016402	112265	000061		MOVB	(R2)+,DR+1(R5)	;SAVE UNIT SUMMARY		
6587	016406	005202			INC	R2	;GET TO WORD BOUND		
6588	016410	012237	016772		MOV	(R2)+,PDTFLG	;AND GET PDT FLAG		
6589	016414	052715	040000		BIS	#BIT14,R5	;SET SEND BREAK FLAG		
6590	016420	032765	000400	000060	BIT	#BIT8,DR(R5)	;DRIVE 0?		
6591	016426	001011			BNE	13	;YES		
6592	016430	032765	001000	000060	BIT	#BIT9,DR(R5)	;DRIVE 1?		
6593	016436	001005			BNE	13	;OK		
6594	016440				ERRSF	102.,NODRVS	;NEITHER?!		
(4)	016440	104454						TRAP	C\$ERSF
(5)	016442	000146						.WORD	102
(5)	016444	016736						.WORD	NODRVS
(5)	016446	000000						.WORD	0
6595	016450								
(3)	016450	104444			DOCLN		;EXIT	TRAP	C\$DCLN
6596									
6597	016452	105737	016770	13:	TSTB	STRT	;START COMMAND?		
6598	016456	001412			BEQ	14	;NO, DONT CLEAR		
6599							;YES-CLEAR STATS		
6600	016460	012702	000202		MOV	#BLKEND,R2	;R2-->END OF STATS		
6601	016464	012701	000110		MOV	#WRTNO,R1	;FORM ADDRESS OF START:		
6602	016470	060501			ADD	R5,R1	;R1-->START OF STATS.		
6603	016472	162702	000110		SUB	#WRTNO,R2	;FORM # TO CLEAR		
6604									
6605	016476	105021		2:	CLRB	(R1)+	;CLEAR 'EM		
6606	016500	005302			DEC	R2	;MORE?		
6607	016502	001375			BNE	2	;YES		
6608	016504	042715	100000	14:	BIC	#BIT15,R5	;SET NOT ABORTED		
6609	016510	010365	000022		MOV	R3,RCSR(R5)	;GET DEVICE REGISTERS:		
6610	016514	062703	000002		ADD	#2,R3			
6611	016520	010365	000024		MOV	R3,RCDB(R5)			
6612	016524	062703	000002		ADD	#2,R3			
6613	016530	010365	000026		MOV	R3,XMSR(R5)			
6614	016534	062703	000002		ADD	#2,R3			
6615	016540	105737	016772		TSTB	PDTFLG	;UNIT A PDT?		
6616	016544	001402			BEQ	4	;NO		
6617	016546	162703	000004		SUB	#4,R3	;YES...RCDB=XMDB		
6618	016552	010365	000000	4:	MOV	R3,XMDB(R5)			
6619	016556	005065	000002		CLR	PATTEN(R5)	;ZERO DATA PATTERN		
6620	016562	005065	000002		CLR	RETRY(R5)	;NO RETRIES		
6621	016566	005065	000004		CLR	REC(R5)	;NO RECORD		
6622	016572	005065	000007		CLR	SUCCS(R5)	;NO SUCCESS		
6623	016576	005065	000007		CLR	DLV(R5)	;NO DLV ERROR		
6624	016602	005065	000210		CLR	MRSP(R5)	***** CLR MRSP INDICATOR		
6625	016606	005037	003342		CLR	ALLGON	;OK TO PRINT STATISTICS		
6626	016612	062737	000002	003314	ADD	#2,DEVPTR	;-->NEXT DEVICE		
6627	016620	005204			INC	R4	;INCREMENT UNIT NUMBER		
6628	016622	020437	002012		CMP	R4,L#UNIT	;MORE UNITS?		
6629	016626	001250			BNE	1	;YES, GP HARD THE NEXT		
6630									
6631	016630	005037	003310		CLR	SYSTAT	;SYSTEM STATUS WORD		
6632	016634				RFLAGS	FLGLQC	;GET USER FLAGS		
(3)	016634	104421						TRAP	C\$RFLA


```

(3) 016636 010037 016774
6633 016642 005037 003334
6634 016646 013737 002210 003312
6635 016654 006237 003312
6636 016660 012737 000200 003336
6637 016666 022737 000200 003312
6638 016674 101003
6639 016676 012737 000400 003336
6649
6661
6662 016704
(3) 016704
(3) 016704 104411
6663
6664
6665 016706 047524 020117 040515
6666 016736
6667 016736 042523 042514 052103
6668 016770
6669 016770 000000
6670 016772 000000
6671 016774 000000

5$: CLR BLKER ;NO ERROR
SETLEN: MOV LENGTH,TAPLEN ;GET # OF RECORDS
ASR TAPLEN ;GET # BLOCKS PER TRACK
MOV #200,SECREC ;PRESET SECOND START AT 200
CMP #200,TAPLEN ;# BLKS > 128.?
BHI 3$ ;NO-SWITCH TRACKS 2ND PASS
MOV #400,SECREC ;YES-START AT 400

3$: ENDINIT

L10007: TRAP C$INIT

TOMANY: .ASCIZ /TOO MANY UNITS MAX.=8 /
.EVEN
NODRVS: .ASCIZ /SELECT AT LEAST 1 DRIVE /
.EVEN
STRT: .WORD
PDTFLG: .WORD ;TUS8 IS IN PDT
FLGLOC: .WORD ;USER FLAGS

```

```

6674
6675
6676
6677
6678
6679 016776          BGNAUTO
(3) 016776
6680 016776 000240          NOP          ;AUTO DROP ROUTINE          L$AUTO::
6681 017000          SETVEC #4,#TRPHND,#PRI07          ;GET BUS TRAP VEC.
(7) 017000 012746 000340          MOV          #PRI07,-
(6) 017004 012746 017106          MOV          #TRPHND,
(5) 017010 012746 000004          MOV          #4,-(SP)
(4) 017014 012746 000003          MOV          #3,-(SP)
(3) 017020 104437          TRAP        C$SVEC
(2) 017022 062706 000010          ADD          #10,SP
6682 017026 012737 003352 017104          MOV          #BLKTBL,TRPPTR ;GET TOP OF DATA BLOCK TABLE
6683 017034 017705 000044          1$: MOV          #TRPPTR,R5 ;GET DATA BLOCK
6684 017040 032715 104000          BIT          #BIT15:BIT11,#R5 ;NOT TESTED OR ABORTED?
6685 017044 100400          BMI          2$ ;YES
6686 017046 005775 000022          TST          #RCSR(R5) ;NO-VALID ADDRESS?
6687 017052 000240          NOP          ;YES...(TRAP IF NOT)
6688 017054 023727 017104 003370          2$: CMP          TRPPTR,#LSTDEV ;MORE TO TRY?
6689 017062 103004          BHS          3$ ;NO
6690 017064 062737 000002 017104          ADD          #2,TRPPTR ;ON TO NEXT
6691 017072 000760          BR           1$ ;GET IT
6692 017074          3$: CLRVEC #4 ;RESTORE
(3) 017074 012700 000004          MOV          #4,R0
(3) 017100 104436          TRAP        C$CVEC
6693 017102          ENDAUTO
(3) 017102
(3) 017104 104461          L10010: TRAP        C$AUTO
6694 017104 000000          TRPPTR: .WORD
6695
6696
6697
6698
6699
6700          ;ILLEGAL ADDRESS TRAP HANDLER:
6701 017106          TRPHND: PRINTF #MSAUTO          ;SAY "AUTO DROPPED"
(7) 017106 012746 017140          MOV          #MSAUTO,
(6) 017112 012746 000001          MOV          #1,-(SP)
(3) 017116 010600          MOV          SP,R0
(4) 017120 104417          TRAP        C$PNT=
(4) 017122 062706 000004          ADD          #4,SP
6702 017126 011500          MOV          #R5,R0 ;GET UNIT #
6703 017130 042700 177770          BIC          #177770,R0 ;MASK IT OFF
6704 017134          DODU        R0 ;DROP HIM
(3) 017134 104451          TRAP        C$DODU
6705 017136 000002
6706 017140 040445 052501 047524          RTI
MSAUTO: .ASCIZ /#AAUTO DROP; #N/

```

6709
6710
6711
6712
6713
6714
6715
6716
(3)
6717
6718
6719
6720
6721
(3)
6722
6729
6741
6742
(3)
6743
(3)
(3)

017160
017160
017160 005737 003342
017164 001004
017166 005737 002212
017172 001401
017174 104424

017176
017176 104433
017200
017200
017200 104412

.SBTTL CLEANUP CODING SECTION

; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.

BGNCLN

TST ALLGON
BNE 1\$
TST STAEOP
BEQ 1\$
DORPT

L\$CLEAN:;
; ENTRANCE FROM ALL-UNITS-ABORTED?
; YES-EXIT
; NO-STATS AT EOP?
; NO
; YES

TRAP C\$DRPT

1\$: BRESET

ENDCLN

TRAP C\$RESET

L10011:

TRAP C\$CLEAN

```

5746                                     .SBTTL  DROP UNIT SECTION
6747
6748
6749                                     ;**
6750                                     ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
6751                                     ; TO NO LONGER BE TESTED.
6752                                     ;**
6753 017202                                BGNDU
6754 (3) 017202                                L#DU:
6755 017202                                ;RO=UNIT NUMBER
6756 (1) 017202 010046                       PUSH  R0          ;SAVE IT
6757 (1)                                MOV   R0,-(SP)
6758 (1)
6759 017204                                ;SAVE PRESENT UNIT POINTER
6760 (1) 017204 010546                       PUSH  R5          MOV   R5,-(SP)
6761 (1)
6762 (1)
6763 017206 004737 017246                   CALL  GETR5       ;GET POINTER TO UNIT
6764 017212 052715 100000                   BIS   #BIT15,R5  ;SET ABORTED
6765 017216 012605                       POP   R5         ;RESTORE PRESENT UNIT POINTER
6766 (1) 017216 012605                       MOV   (SP)+,R5
6767 (1)
6768 017220                                ;RETRIEVE UNIT NUMBER
6769 (1) 017220 012600                       POP   R0         MOV   (SP)+,R0
6770 (1)
6771 017222                                PRINTF #ABOMSG,R0
6772 (8) 017222 010046
6773 (7) 017224 012746 017300
6774 (6) 017230 012746 000002
6775 (3) 017234 010600
6776 (4) 017236 104417
6777 (4) 017240 062706 000006
6778
6779
6780
6781 017244                                ENDCU
6782 (3) 017244
6783 (3) 017244 104453                                L10012:
6784 017246 012737 003352 017276           GETR5: MOV   #BLKTBL, PTR
6785 017254 017705 000016           1#;   MOV   #PTR,R5
6786 017260 005300                       DEC   R0
6787 017262 100404                       BMI   2#
6788 017264 062737 000002 017276           ADD   #2,PTR
6789 017272 000770                       BR    1#
6790 017274 000207           2#;   RETURN
6791 017276 000000           PTR:  .WORD
6792
6793 017300 040445 051104 050117           ABOMSG: .ASCIZ /#ADROPPED UNIT #D1#N/
6794 017326                                .EVEN

```

C8

MISCELLANEOUS SECTIONS MACY11 30(1046) 25-JAN-84 08:33 PAGE 50
CZTUUF.P11 25-JAN-84 08:09 ADD UNIT SECTION

SEQ 0093

6795
6796
6797
6798
6799
6800
6801
6802
6803
6804
6805
6811
6823
6824
6825
6826
6827

017326
(3) 017326
017326
(3) 017326
(3) 017326 104452

.SBTTL ADD UNIT SECTION

; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
; TO THE TEST CYCLE.

BGNAU

L\$AU::

;THE INIT CODE CONTAINS ALL CODE NECESSARY TO ADD A UNIT.

ENDAU

L10013:

TRAP

C\$AU

```

6886          .SBTTL TEST 1 / DEVICE SELF-DIAGNOSTIC EXECUTION
6887
6888 017330
6889
6890
6891 017330
6892 017330
6893 017370
6894
6895 017374
6896 017522
6897 017526
6898
6899
6900 017530
6901 017530
6902 017530

          .NLIST  BGNMOD
          ME,BEX

          BGNTST

          TSTID  #TST1

          MOV      #TST1,TSTTOP      ;SAVE ADDR OF TEST
          CALL    SETUP              ;INIT UNITS TSTPC
          CALL    SETDR              ;GET 1ST DRVS.
          CALL    RUN                ;DO TEST
          CALL    SWAPDR             ;GET NEXT DRVS.
          BCC     64$                ;BR NO 2ND DRVS
          CALL    SETUP              ;REINIT UNITS TSTPC
          CALL    RUN                ;REPEAT TEST
          ;DONE

          EXIT TST                  64$:

          TRAP    C$EXIT
          .WORD   L10014-.

TST1:  TUSELF

          64$:  MOV      #TRBUF,R0      ;FORM COMMAND PACKET
          MOVB   #RSCMND,BR0         ;COMMAND FLAG
          MOVB   #RSMSIZ,1(R0)      ;SIZE OF MESSAGE
          MOVB   #RSSSLF,2(R0)      ;SELF TEST OPERATION
          CLRB   3(R0)              ;NO MODIFIER.
          CLR    4(R0)              ;NO DRIVE OR SWITCHES
          CLR    6(R0)              ;NO SEQUENCE NUMBER
          CLR    8.(R0)             ;NO BYTES
          CLR    10.(R0)            ;NO RECORD #
          MOV    #RSMSIZ,R1         ;GET SIZE
          TST   (R1),               ;+2 FOR CHECKSUM
          MOV    #RSSNSZ,SND CNT(R5) ;SIZE TO SEND
          CALL   CHKSUM             ;FORM CHECKSUM
          MOV    R1,(R0)            ;INSERT INTO PACKET
          MOV    #RSEND,XSFLG(R5)   ;EXPECT END.
          MOV    #RSNDSZ,XSCNT(R5) ;THIS BIG
          MOV    #1,XSPKNT(R5)     ;AND 1 PACKET
          ;SEND
          CALL   RSVP              ;RETURN TO SCHEDULER
          BIT    #BIT3,R5          ;RETRY?(BAD FLAG)
          BNE   64$                ;YES

          INC    DONE
          RETURN

          ENDTST

          L10014:  TRAP    C$ETST

```

```

6903          .SBTTL TEST 2 / SEEK EOT,BOT
6904
6905          BGNTST
(3) 017532
6906          TSTID #TST2          T2::
(1) 017532 012737 017576 003330          MOV #TST2,TSTTOP          ;SAVE ADDR OF TEST
(1) 017540 004737 006024          CALL SETUP          ;INIT UNITS TSTPC
(1) 017544 004737 005652          CALL SETDR          ;GET 1ST DRVS.
(1) 017550 004737 006072          CALL RUN          ;DO TEST
(1) 017554 004737 005532          CALL SWAPDR          ;GET NEXT DRVS.
(1) 017560 103004          BCC 64$          ;BR NO 2ND DRVS
(1) 017562 004737 006024          CALL SETUP          ;REINIT UNITS TSTPC
(1) 017566 004737 006072          CALL RUN          ;REPEAT TEST
(1) 017572          ;DONE
6907          EXIT TST          64$:
(3) 017572 104432          TRAP C$EXIT
(3) 017574 000206          .WORD L10015-.
6908
6909
6910          TST2: CLR R4          ;R4=INDEX INTO RECORD TABLE
6911          1$: MOV RECDAT(R4),REC(R5) ;GET THE RECORD
6912
6913          TUSEEK REC(R5),DR(R5) ;SEEK IT
(1)
(1) 017606 012700 027746          64$: MOV #TRBUF,R0          ;-->(POINT TO) XMIT BUFF
(1) 017612 112710 000002          MOVB #RSCMND,BR0          ;FORM COMMAND MESSAGE PA
(1) 017616 112760 000012 000001          MOVB #RSMISZ,1(R0)          ;THIS BIG
(1) 017624 112760 000005 000002          MOVB #RSSSEK,2(R0)          ;OP CODE IS SEEK
(1) 017632 016560 000064 000012          MOV REC(R5),10.(R0)          ;TO THIS RECORD
(1) 017640 116560 000060 000004          MOVB DR(R5),4.(R0)          ;AND WHICH DRIVE
(1) 017646 105060 000003          CLRB 3.(R0)          ;NO MODIFIER
(1) 017652 105060 000005          CLRB 5.(R0)          ;NO SWITCHES
(1) 017656 005060 000006          CLR 6.(R0)          ;NO SEQUENCE #
(1) 017662 005060 000010          CLR 8.(R0)          ;NO BYTE COUNT
(1) 017666 012701 000012          MOV #RSMISZ,R1          ;GET COUNT
(1) 017672 005721          TST (R1)+          ;PLUS FLAG + BCNT
(1)          ;FOR CHECKSUM CALC
(1) 017674 004737 013674          CALL CHKSUM          ;R0-->TOP R1=# OF BYTE
(1) 017700 010110          MOV R1,(R0)          ;INSERT INTO PACKET
(1)          ;SET UP EXPECTATIONS:
(1) 017702 012765 000016 000070          MOV #RSSNSZ,SNDCNT(R5) ;HOW MANY TO SEND
(1) 017710 112765 000002 000034          MOVB #RSCMND,XSFLG(R5) ;EXPECT END PACK
(1) 017716 012765 000016 000036          MOV #RSNDSZ,XSCNT(R5) ;COUNT WITH THIS
(1) 017724 012765 000001 000032          MOV #1.,XSPKNM(R5) ;EXPECT ONLY 1 PACKET
(1)
(1) 017732 004737 006600          CALL RSVF          ;SEND
(1)          ;AND RETURN TO SCHEDULER
(1) 017736 032715 000010          BIT #BIT3,DR5          ;RETRY (FLAG BYTE ERROR)
(1) 017742 001321          BNE 64$          ;YES
6914
6915          ADD #2,R4          ;POINT TO NEXT RECORD
6916          CMP RECDAT(R4),#-1,          ;LAST ONE DONE?
6917          BNE 1$          ;NO-LOOP
6918          INC DONE          ;YES-SET DONE FLAG
6919          RETURN

```

F8

MISCELLANEOUS SECTIONS MACY11 30(1046) 25-JAN-84 08:33 PAGE 52-1
CZTUUF.P11 25-JAN-84 08:09 TEST 2 / SELK EOT,BOT

SEQ 0096

6920
6921 017766 000000
6922 017770 000200
6923 017772 000177
6924 017774 000377
6925 017776 000400
6926 020000 177777
6927 020002
(3) 020002
(3) 020002 104401

RECDAT: 0. ;BOT
200 ;BOT OTHER TRACK
177 ;EOT
377 ;EOT OTHER TRACK
400 ;BOT AGAIN
-1.
ENDTST

L10015: TRAP C\$ETST

.SBTTL TEST 3 / HIGH ACTIVITY WRITE/READ (512 BYTE/BLOCK MODE)

; WRITE THEN READ VARYING DATA FOR ALL PHYSICALLY ADJACENT BLOCKS AROUND
; A RECORD, GO HALF-WAY INTO REMAINING TAPE REPEAT UNTIL EOT.

```

6930
6931
6932
6933
6934
6935 020004          BGNTST
(3) 020004
6936 020004          TSTID  @TST3
(1) 020004 012737 020050 003330      MOV    @TST3,TSTTOP      ;SAVE ADDR OF TEST
(1) 020012 004737 006024              CALL  SETUP             ;INIT UNITS TSTPC
(1) 020016 004737 005652              CALL  SETDR            ;GET 1ST DRVS.
(1) 020022 004737 006072              CALL  RUN              ;DO TEST
(1) 020026 004737 005532              CALL  SWAPDR           ;GET NEXT DRVS.
(1) 020032 103004              BCC   64$             ;BR NO 2ND DRVS
(1) 020034 004737 006024              CALL  SETUP            ;REINIT UNITS TSTPC
(1) 020040 004737 006072              CALL  RUN              ;REPEAT TEST
(1) 020044              EXIT TST          64$:          ;DONE
6937 020044
(3) 020044 104432
(3) 020046 001326          TRAP   C$EXIT
                          .WORD   L10016-.
6938
6939
6940 020050 012765 000100 000066      TST3:  MOV    @100,TMP(R5)    ;INIT TO HALF OF REMAINING
6941 020056 005004              CLR    R4              ;FOR INDEX INTO DATA TABLE
6942 020060 005065 000064              CLR    REC(R5)        ;START AT RECORD 0
6943 020064 016465 022766 000072      1$:   MOV    TST3PT(R4),PATTEN(R5) ;GET DATA
6944 020072              TUWRIT PATTEN(R5),REC(R5),@512.,DR(R5),@0
(1) 020072 012700 027744              72$:  MOV    @TRBUF,R0      ;MAKE COMMAND PACKET:
(1) 020076 112710 000002              MOVB  @RSCMND,@R0     ;COMMAND FLAG
(1) 020102 112760 000012 000001      MOVB  @RSMSTZ,1(R0)   ;THIS SIZE
(1) 020110 112760 000003 000002      MOVB  @RSSWR,2(R0)    ;INSERT OP CODE-WRITE
(1) 020116 112760 000000 000003      MOVB  @0,3.(R0)       ;VERIFY (1 OR 0)
(1) 020124 116560 000060 000004      MOVB  DR(R5),4.(R0)   ;DRIVE #
(1) 020132 112760 000020 000005      MOVB  @020,5.(R0)    ;MAINTENANCE MODE SWITCH
(1) 020140 005060 000006              CLR    6.(R0)         ;NO SEQUENCE #
(1) 020144 012760 001000 000010      MOV    @512.,8.(R0)   ;TOTAL COUNT TO WRITE
(1) 020152 016560 000064 000012      MOV    REC(R5),10.(R0);AT RECORD N
(1) 020160 012701 000012              MOV    @RSMSTZ,R1     ;THE PACKET SIZE PLUS+2
(1) 020164 005721              TST   (R1)+           ;(FLAG AND COUNT) INTO R
(1) 020166 012765 000016 000070      MOV    @RSSNSZ,SNOCNT(R5);LOAD THE SIZE TO S
(1) 020174 004737 013674              CALL  CHKSUM          ;R0 --> R1-COUNT
(1) 020200 010110              MOV    R1,(R0)        ;PUT CHKSUM IN PACKET
(1) 020202 012765 000020 000034      MOV    @RSCONT,XSFLG(R5);THE FLAG
(1) 020210 012765 000001 000036      MOV    @1,XSCNT(R5)   ;THE COUNT
(1) 020216 012765 000001 000032      MOV    @1,XSPKNT(R5)  ;THE # PACKETS EXPECTED
(1) 020224 012702 001000              MOV    @512.,R2      ;GET # OF DATA B
(1) 020230 004737 006600              CALL  RSVP            ;SEND (AND RETURN TO SCH
(1) 020234 032715 000010              BIT   @BIT3,@R5      ;FLAG BYTE ERROR?
(1) 020240 001314              BNE   72$            ;YES
(1) 020242 042715 010000              BIC   @BIT12,@R5     ;FLAG FOR LAST PACKET
(1) 020246 012700 027746              64$:  MOV    @TRBUF,R0      ;POINT TO TOP OF BUFFER
(1) 020252 020227 000200              CMP   R2,@128.       ;START DATA PACKET(S)
(1) 020256 101004              BHI   65$            ;@512. > 128.!
(1) 020260 010201              MOV   R2,R1          ;@512.<128.
(1) 020262 052715 010000              BIS   @BIT12,@R5     ;SO LAST PACKET NOW

```

(1)	020266	000402				BR	66\$; USE REMAINING COUNT
(1)	020270	012701	000200			65\$: MOV	#128.,R1		; USE 128. BYTES
(1)	020274	110160	000001			66\$: MOVB	R1,1(R0)		; COPY COUNT TO BUFFER
(1)	020300	010103					MOV	R1,R3	; R3=COUNTER TO LOAD BUFF
(1)	020302	112710	000001				MOVB	#RSDATA,#R0	; FLAG FIRST
(1)	020306	005720					TST	(R0)+	; SKIP COUNT
(1)	020310	116520	000072			67\$: MOV	PATTEN(R5),(R0)+		; INSERT DATA
(1)	020314	005303					DEC	R3	; MORE?
(1)	020316	101374					BHI	67\$; YES
(1)	020320	012700	027746				MOV	#TRBUF,R0	; -->TOP AGAIN
(1)	020324	116001	000001				MOVB	1(R0),R1	; GET COUNT
(1)	020330	042701	177400				BIC	#177400,R1	; ZERO SIGN EXTEND
(1)	020334	010165	000070				MOV	R1,SNDCNT(R5)	; HOW MANY TO SEND PLUS
(1)	020340	062765	000004	000070			ADD	#4,SNDCNT(R5)	; FLAG,COUNT,CHKSUM
(1)	020346	062701	000002				ADD	#2,R1	; COMPENSATE FOR FLAG + C
(1)	020352	004737	013674				CALL	CHKSUM	; FOR CHECKSUM CALC.
(1)	020356	110120					MOVB	R1,(R0)+	; CHKSUM INTO PACKET
(1)	020360	000301					SWAB	R1	; EVEN ON AN ODD
(1)	020362	110120					MOVB	R1,(R0)+	; BYTE BOUNDARY
(1)	020364	032715	010000				BIT	#BIT12,#R5	; LAST DATA PACKET?
(1)	020370	001412					BEQ	68\$; NO
(1)	020372	012765	000002	000034			MOV	#RSEND,XSFLG(R5)	; YES-EXPECT 'END'
(1)	020400	012765	000016	000036			MOV	#RSSNSZ,XSCNT(R5)	; OF THIS SIZE
(1)	020406	012765	000001	000032			MOV	#1,XSPKNT(R5)	; AND 1 PACKET
(1)	020414	000411					BR	69\$; SEND
(1)	020416	012765	000020	000034		68\$: MOV	#RSCONT,XSFLG(R5)		; (NOT LAST), EXPECT
(1)	020424	012765	000001	000036			MOV	#1,XSCNT(R5)	; AND 1 BYTE
(1)	020432	012765	000001	000032			MOV	#1,XSPKNT(R5)	; AND 1 PACKET
(1)	020440	004737	006600			69\$: CALL	RSVP		; SEND PACKET
(1)									; AND RETURN TO SCHEDULER
(1)	020444	032715	000010				BIT	#BIT3,#R5	; FLAG BYTE RETRY?
(1)	020450	001210					BNE	72\$; YES
(1)	020452	032715	002000				BIT	#BIT10,#R5	; RETRY DATA ERROR?
(1)	020456	001004					BNE	70\$; YES
(1)	020460	162702	000200				SUB	#128.,R2	; NO, MORE DATA TO SEND?
(1)	020464	101270					BHI	64\$; YES
(1)	020466	000502					BR	71\$; NO
(1)	020470					70\$: TURTRY	REC(R5),#512.,DR(R5)		; RETRY HERE
(2)									
(2)									
(2)	020470	012700	027746			76\$: MOV	#TRBUF,R0		; FORM CMD PACK:
(2)	020474	112710	000002				MOVB	#RSCMD,#R0	; MESSAGE PACK TYPE
(2)	020500	112760	000012	000001			MOVB	#RMSIZ,2(R0)	; THIS BIG
(2)	020506	112760	000002	000002			MOVB	#RSSRD,.(R0)	; OP CODE-READ
(2)	020514	016560	000064	000012			MOV	REC(R5),10.(R0)	; THIS RECORD
(2)	020522	116560	000060	000004			MOVB	DR(R5),4.(R0)	; THIS DRIVE
(2)	020530	105000	000003				CLRB	3(R0)	; PRESET NORM THRESHOLD
(2)	020534	105715					TSTB	#R5	; REDUCED?
(2)	020536	100002					BPL	77\$; NO
(2)	020540	105260	000003				INCB	3(R0)	; YES-CHANGE THRESHOLD
(2)	020544	012750	001000	000010		77\$: MOV	#512.,8.(R0)		; # BYTES DESIRED
(2)	020552	112760	000020	000005			MOVB	#020,5.(R0)	; MAINTENANCE MODE
(2)	020560	005060	000006				CLR	6.(R0)	; NO SEQUENCE #
(2)	020564	012701	000012				MOV	#RMSIZ,R1	; SIZE OF PACKET
(2)	020570	005721					TST	(R1)+	; PLUS FLAG,COUNT INTO R1
(2)	020572	012765	000016	000070			MOV	#RSSNSZ,SNDCNT(R5)	; SET UP SIZE TO SEND

MISCELLANEOUS SECTIONS MACY11 30(1046) 25-JAN-84 08:33 PAGE 53-2
CZTUUF.P11 25-JAN-84 08:09

TEST 3 / HIGH ACTIVITY WRITE/READ (512 BYTE/BLOCK MODE)

SEQ 0099

(2)											
(2)	020600	004737	013674						CALL	CHKSUM	;FORM CHECKSUM R1=COUNT
(2)	020604	010110							MOV	R1,(R0)	;INSERT IN PACKET
(2)											
(2)	020606	012701	001000						MOV	#512.,R1	;SET EXPECTATION
(2)											;CALC # OF DATA PACKETS
(2)	020612	012703	000034						MOV	#XSFLG,R3	;OFFSET OF FLAG
(2)	020616	060503							ADD	R5,R3	;ABS. ADDR. OF XSFLG
(2)	020620	005002							CLR	R2	;PRESET
(2)	020622	005202						73:	INC	R2	;# PACKETS EXPECTED
(2)	020624	012723	000001						MOV	#RSDATA,(R3)+	;LOAD XSFLG
(2)	020630	012723	000204						MOV	#132.,(R3)+	;AND EXPECT COUNT
(2)	020634	162701	000200						SUB	#128.,R1	;NEG RESULT LAST TIME
(2)	020640	101401							BLOS	75:	;LAST TIME!
(2)	020642	000767							BR	73:	;MORE TO DO
(2)	020644	005202						75:	INC	R2	;ADD ONE FOR END PACK
(2)	020646	010265	000032						MOV	R2,XSPKMM(R5)	;SAVE # PACKETS TO EXPEC
(2)	020652	012723	000002						MOV	#RSEND,(R3)+	;EXPECT AN END
(2)	020656	012713	000016						MOV	#RSNDSZ,(R3)	;THIS BIG-14. BYTES
(2)											
(2)	020662	004737	006600						CALL	RSVP	;SEND
(2)											;AND RETURN TO SCHEDULER
(2)											
6945	020676								TUREAD	REC(R5),#512.,DR(R5),#0	
(1)											
(1)											
(1)	020675	012700	027746					82:	MOV	#TRBUF,R0	;FORM CMD PACK;
(1)	020702	112710	000002						MOV	#RSCMD,#R0	;MESSAGE PACK TYPE
(1)	020706	112760	000012	000001					MOV	#RSMSIZ,1(R0)	;THIS BIG
(1)	020714	112760	000002	000002					MOV	#RSSRD,2(R0)	;OP CODE IS READ
(1)	020722	016560	000064	000012					MOV	REC(R5),10.(R0)	;THIS RECURD
(1)	020730	116560	000060	000004					MOV	DR(R5),4.(R0)	;THIS DRIVE
(1)	020736	112760	000000	000003					MOV	#0,3.(R0)	;VERIFY
(1)	020744	012760	001000	000010					MOV	#512.,8.(R0)	;TOTAL BYTES TO READ
(1)	020752	112760	000020	000005					MOV	#020,5.(R0)	;MAINTENANCE MODE
(1)	020760	005060	000006						CLR	6.(R0)	;NO SEQUENCE #
(1)	020764	012701	000012						MOV	#RSMSIZ,R1	;GET SIZE OF PACKET
(1)	020770	005721							TST	(R1)+	;+2 FOR CHECKSUM
(1)	020772	012765	000016	000070					MOV	#RSSNSZ,SNDCNT(R5)	;SIZE TO SEND
(1)	021000	004737	013674						CALL	CHKSUM	;FORM CHECKSUM R1=COUNT
(1)	021004	010110							MOV	R1,(R0)	;INSERT CHECKSUM
(1)											
(1)	021006	012701	001000						MOV	#512.,R1	;SET EXPECTATION
(1)											;CALC # OF DATA PACKETS
(1)	021012	012703	000034						MOV	#XSFLG,R3	;GET OFFSET
(1)	021016	060503							ADD	R5,R3	;ABS. ADDR. OF XSFLG
(1)	021020	005002							CLR	R2	;PRESET AS NONE
(1)	021022	005202						78:	INC	R2	;# PACKETS EXPECTED
(1)	021024	012723	000001						MOV	#RSDATA,(R3)+	;LOAD XSFLG
(1)	021030	012723	000204						MOV	#132.,(R3)+	;AND EXPECTED COUNT
(1)	021034	162701	000200						SUB	#128.,R1	;NEG RESULT LAST TIME
(1)	021040	101401							BLOS	80:	;LAST TIME
(1)	021042	000767							BR	78:	;MORE TO DO
(1)	021044	005202						80:	INC	R2	;ADD ONE FOR END PACK
(1)	021046	010265	000032						MOV	R2,XSPKMM(R5)	;SAVE # PACKETS TO EXPEC
(1)	021052	012723	000002						MOV	#RSEND,(R3)+	;EXPECT AN END ALSO...

(1)	021056	012713	000016			MOV	0RSNDSZ,(R3)	;THIS BIG-14. BYTES
(1)	021062	004737	006600			CALL	RSVP	;SEND
(1)								;AND RETURN TO SCHEDULER
(1)	021066	032715	002010			81\$:	BIT	0BIT10!BIT3,0R5 ;RETRY?
(1)	021072	001500					BEQ	79\$;NO.
(1)	021074						TURTRY	REC(R5),0512.,DR(R5) ;YES
(2)								
(2)								
(2)	021074	012700	027746			86\$:	MOV	0TRBUF,R0 ;FORM CMND PACK;
(2)	021100	112710	000002				MOVB	0RSCMND,0R0 ;MESSAGE PACK TYPE
(2)	021104	112760	000012	000001			MOVB	0RSMSIZ,1(R0) ;THIS BIG
(2)	021112	112760	000002	000002			MOVB	0RSSRD,2(R0) ;OP CODE-READ
(2)	021120	016560	000064	000012			MOV	REC(R5),10.(R0) ;THIS RECORD
(2)	021126	116560	000060	000004			MOVB	DR(R5),4.(R0) ;THIS DRIVE
(2)	021134	105060	000003				CLRB	3(R0) ;PRESET NORM THRESHOLD
(2)	021140	105715					TSTB	0R5 ;REDUCED?
(2)	021142	100002					BPL	87\$;NO
(2)	021144	105260	000003				INCB	3(R0) ;YES-CHANGE THRESHOLD
(2)	021150	012760	001000	000010		87\$:	MOV	0512.,8.(R0) ;0 BYTES DESIRED
(2)	021156	112760	000020	000005			MOVB	0020,5.(R0) ;MAINTENANCE MODE
(2)	021164	005060	000006				CLR	6.(R0) ;NO SEQUENCE 0
(2)	021170	012701	000012				MOV	0RSMSIZ,R1 ;SIZE OF PACKET
(2)	021174	005721					TST	(R1)+ ;PLUS FLAG+COUNT INTO R1
(2)	021176	012765	000016	000070			MOV	0RSSNSZ,SNDCNT(R5) ;SET UP SIZE TO SEND
(2)								
(2)	021204	004737	013674				CALL	CHKSUM ;FORM CHECKSUM R1=COUNT
(2)	021210	010110					MOV	R1,(R0) ;INSERT IN PACKET
(2)								
(2)	021212	012701	001000				MOV	0512.,R1 ;SET EXPECTATION
(2)								;CALC 0 OF DATA PACKETS
(2)	021216	012703	000034				MOV	0XSFLG,R3 ;OFFSET OF FLAG
(2)	021222	060503					ADD	R5,R3 ;ABS. ADDR. OF XSFLG
(2)	021224	005002					CLR	R2 ;PRESET
(2)	021226	005202				83\$:	INC	R2 ;0 PACKETS EXPECTED
(2)	021230	012723	000001				MOV	0RSDATA,(R3)+ ;LOAD XSFLG
(2)	021234	012723	000204				MOV	0132.,(R3)+ ;AND EXPECT COUNT
(2)	021240	162701	000200				SUB	0128.,R1 ;NEG RESULT LAST TIME
(2)	021244	101401					BLOS	85\$;LAST TIME!
(2)	021246	000767					BR	83\$;MORE TO DO
(2)	021250	005202				85\$:	INC	R2 ;ADD ONE FOR END PACK
(2)	021252	010265	000032				MOV	R2,XSPKNM(R5) ;SAVE 0 PACKETS TO EXPECT
(2)	021256	012723	000002				MOV	0RSEND,(R3)+ ;EXPECT AN END
(2)	021262	012713	000016				MOV	0RSNDSZ,(R3) ;THIS BIG-14. BYTES
(2)								
(2)	021266	004737	006600				CALL	RSVP ;SEND
(2)								;AND RETURN TO SCHEDULER
(2)								
6946	021276	062704	000002				ADD	02,R4 ;POINT TO NEXT DATA
6947	021302	005764	022766				TST	TST3PT(R4) ;END?
6948	021306	001402					BEQ	2\$;YES
6949	021310	000137	020064				JMP	1\$;NO-WRITE, READ NEW DATA
6950	021314	005004				2\$:	CLR	R4 ;POINT TO FIRST DATA
6951	021316	062765	000200	000064			ADD	0200,REC(R5) ;BUT NOW USE ADJACENT RECORD
6952	021324	032765	001000	000064			BIT	01000,REC(R5) ;ALL ADJACENT RECORDS DONE?
6953	021332	001002					BNE	3\$;YES
6954	021334	000137	020064				JMP	1\$;NO-WRITE, READ AT NEW RECORD

MISCELLANEOUS SECTIONS MACY11 30(1046)
CZTUUF.P11 25-JAN-84 08:09

25-JAN-84 08:33 PAGE 53-4
TEST 3 / HIGH ACTIVITY WRITE/READ (512 BYTE/BLOCK MODE)

SEQ 0101

6955	021340	162765	001000	000064	3\$:	SUB	#1000,REC(R5)	;RESTORE TO NEXT RECORD
6956	021346	066565	000066	000064		ADD	TMP(R5),REC(R5)	;HALF INTO REST OF TAPE
6957	021354	006265	000066			ASR	TMP(R5)	;HALF OF HALF FOR NEXT TIME
6958	021360	103402				BCS	4\$;DONE?
6959	021362	000137	020064			JMP	1\$;NO
6960	021366	005237	003324		4\$:	INC	DONE	;YES-SET FLAG
6961	021372	000207				RETURN		
6962	021374					ENDTST		
(3)	021374							
(3)	021374	104401						
6963								

L10016: TRAP C\$ETST

.SBTTL TEST 4 / HIGH ACTIVITY WRITE/READ (128 BYTE/BLOCK MODE)

; WRITE THEN READ VARYING DATA FOR ALL PHYSICALLY ADJACENT BLOCKS AROUND
; A RECORD, GO HALF-WAY INTO REMAINING TAPE REPEAT UNTIL EOT.

6965
6966
6967
6968
6969
6970 021376
(3) 021376
6971 021376
(1) 021376 012737 021442 003330
(1) 021404 004737 006024
(1) 021410 004737 005652
(1) 021414 004737 006072
(1) 021420 004737 005532
(1) 021424 103004
(1) 021426 004737 006024
(1) 021432 004737 006072
(1) 021436
6972 021436
(3) 021436 104432
(3) 021440 001340
6973
6974
6975 021442 012765 000400 000066
6976 021450 005004
6977 021452 005065 000064
6978 021456 016465 022766 000072
6979 021464
(1) 021464 012700 027746
(1) 021470 112710 000002
(1) 021474 112760 000012 000001
(1) 021502 112760 000003 000002
(1) 021510 112760 000200 000003
(1) 021516 116560 000060 000004
(1) 021524 112760 000020 000005
(1) 021532 005060 000006
(1) 021536 012760 000200 000010
(1) 021544 016560 000064 000012
(1) 021552 012701 000012
(1) 021556 005721
(1) 021560 012765 000016 000070
(1) 021566 004737 013074
(1) 021572 010110
(1)
(1) 021574 012765 000020 000034
(1) 021602 012765 000001 000036
(1) 021610 012765 000001 000032
(1) 021616 012702 000200
(1) 021622 004737 006600
(1) 021626 032715 000010
(1) 021632 001314
(1) 021634 042715 010000
(1) 021640 012700 027746
(1) 021644 020227 000200
(1) 021650 101004
(1) 021652 010201
(1) 021654 052715 010000

BGNTST

TSTID #TST4

T4::

MOV #TST4,TSTTOP ;SAVE ADDR OF TEST
CALL SETUP ;INIT UNITS TSTPC
CALL SETDR ;GET 1ST DRVS.
CALL RUN ;DO TEST
CALL SWAPDR ;GET NEXT DRVS.
BCC 64\$;BR NO 2ND DRVS
CALL SETUP ;REINIT UNITS TSTPC
CALL RUN ;REPEAT TEST
;DONE

EXIT TST

64\$:

TRAP C\$EXIT
.WORD L10017-

TST4: MOV #400,TMP(R5) ;INIT TO HALF OF REMAINING
CLR R4 ;FOR INDEX INTO DATA TABLE
CLR REC(R5) ;START AT RECORD 0
1\$: MOV TST3PT(R4),PATTEN(R5) ;GET DATA
TUWRIT PATTEN(R5),REC(R5),#128.,DR(R5),#BIT7

72\$:

MOV #TRBUF,R0 ;MAKE COMMAND PACKET:
MOVB #RSCMND,BR0 ;COMMAND FLAG
MOVB #RSMISZ,1(R0) ;THIS SIZE
MOVB #RSSWR,2(R0) ;INSERT OP CODE-WRITE
MOVB #BIT7,3.(R0) ;VERIFY (1 OR 0)
MOVB DR(R5),4.(R0) ;DRIVE #
MOVB #020.5.(R0) ;MAINTENANCE MODE SWITCH
CLR 6.(R0) ;NO SEQUENCE #
MOV #128.,8.(R0) ;TOTAL COUNT TO WRITE
MOV REC(R5),10.(R0) ;AT RECORD N
MOV #RSMISZ,R1 ;THE PACKET SIZE PLUS+2
TST (R1)+ ;(FLAG AND COUNT) INTO R
MOV #RSSNSZ,SNDCNT(R5) ;LOAD THE SIZE TO S
CALL CHKSUM ;R0 --> R1*COUNT
MOV R1,(R0) ;PUT CHKSUM IN PACKET
;SET UP EXPECTATIONS:
MOV #RSCONT,XSFLG(R5) ;THE FLAG
MOV #1,XSCNT(R5) ;THE COUNT
MOV #1,XSPKNM(R5) ;THE # PACKETS EXPECTED
MOV #128.,R2 ;GET # OF DATA B
CALL RSVP ;SEND (AND RETURN TO SCH
BIT #BIT3,DR5 ;FLAG BYTE ERROR?
BNE 72\$;YES
BIC #BIT12,DR5 ;FLAG FOR LAST PACKET
64\$: MOV #TRBUF,R0 ;POINT TO TOP OF BUFFER
CMP R2,#128. ;START DATA PACKET(S)
BHI 65\$;#128. > 128.!
MOV R2,R1 ;#128. < 128.
BIS #BIT12,DR5 ;SO LAST PACKET NOW

(1)	022450	012713	000016			MOV	#RSNDSZ,(R3)	; THIS BIG-14. BYTES	
(1)	022454	004737	006600			CALL	RSVP	; SEND	
(1)								; AND RETURN TO SCHEDULER	
(1)	022460	032715	002010			811:	BIT	#BIT10!BIT3,DR5	; RETRY?
(1)	022464	001500					BEQ	791	; NO.
(1)	022466						TURTRY	REC(R5),#128.,DR(R5)	; YES
(2)									
(2)									
(2)	022466	012700	027746			861:	MOV	#TRBUF,R0	; FORM CMD PACK:
(2)	022472	112710	000002				MOVB	#RSCHNO,DR0	; MESSAGE PACK TYPE
(2)	022476	112760	000012	000001			MOVB	#RSMSIZ,1(R0)	; THIS BIG
(2)	022504	112760	000002	000002			MOVB	#RSSRD,2(R0)	; OP CODE-READ
(2)	022512	016560	000064	000012			MOV	REC(R5),10.(R0)	; THIS RECORD
(2)	022520	116560	000060	000004			MOVB	DR(R5),4.(R0)	; THIS DRIVE
(2)	022526	105060	000003				CLRB	3(R0)	; PRESET NORM THRESHOLD
(2)	022532	105715					TSTB	DR5	; REDUCED?
(2)	022534	100002					BPL	871	; NO
(2)	022536	105260	000003				INCB	3(R0)	; YES-CHANGE THRESHOLD
(2)	022542	012760	000200	000010		871:	MOV	#128.,8.(R0)	; # BYTES DESIRED
(2)	022550	112760	000020	000005			MOVB	#020,5.(R0)	; MAINTENANCE MODE
(2)	022556	005060	000006				CLR	6.(R0)	; NO SEQUENCE #
(2)	022562	012701	000012				MOV	#RSMSIZ,R1	; SIZE OF PACKET
(2)	022566	005721					TST	(R1)	; PLUS FLAG-COUNT INTO R1
(2)	022570	012765	000016	000070			MOV	#RSSNSZ,SNDCNT(R5)	; SET UP SIZE TO SEND
(2)									
(2)	022576	004737	013674				CALL	CHKSUM	; FORM CHECKSUM R1-COUNT
(2)	022602	010110					MOV	R1.(R0)	; INSERT IN PACKET
(2)									
(2)	022604	012701	000200				MOV	#128.,R1	; SET EXPECTATION
(2)									; CALC # OF DATA PACKETS
(2)	022610	012703	000034				MOV	#XSFLG,R3	; OFFSET OF FLAG
(2)	022614	060503					ADD	R5,R3	; ABS. ADDR. OF XSFLG
(2)	022616	005002					CLR	R2	; PRESET
(2)	022620	005202				831:	INC	R2	; # PACKETS EXPECTED
(2)	022622	012723	000001				MOV	#RSDATA,(R3)	; LOAD XSFLG
(2)	022626	012723	000204				MOV	#132.,(R3)	; AND EXPECT COUNT
(2)	022632	162701	000200				SUB	#128.,R1	; NEG RESULT LAST TIME
(2)	022636	101401					BLOS	851	; LAST TIME!
(2)	022640	000767					BR	831	; MORE TO DO
(2)	022642	005202				851:	INC	R2	; ADD ONE FOR END PACK
(2)	022644	010265	000032				MOV	R2,XSPKNT(R5)	; SAVE # PACKETS TO EXPECT
(2)	022650	012723	000012				MOV	#RSEND,(R3)	; EXPECT AN END
(2)	022654	012713	000016				MOV	#RSNDSZ,(R3)	; THIS BIG-14. BYTES
(2)									
(2)	022660	004737	006600				CALL	RSVP	; SEND
(2)									; AND RETURN TO SCHEDULER
(2)									
6981	022670	062704	000002				ADD	#2,R4	; POINT TO NEXT DATA
6982	022674	005764	022766				TST	TST3PT(R4)	; END?
6983	022700	001462					BEQ	21	; YES
6984	022702	000137	021456				JMP	11	; NO-WRITE, READ NEW DATA
6985	022706	005004				21:	CLR	R4	; POINT TO FIRST DATA
6986	022710	032765	001000	000064			ADD	#1000,REC(R5)	; BUT NOW USE ADJACENT RECORD
6987	022716	032765	001000	000064			BIT	#4000,REC(R5)	; ALL ADJACENT RECORDS DONE?
6988	022724	001002					BNE	31	; YES
6989	022726	000137	021456				JMP	11	; NO-WRITE, READ AT NEW RECORD

MISCELLANEOUS SECTIONS MACY11 30(1046)
CZTUUF.P11 25-JAN-84 08:09

25-JAN-84 08:33 PAGE 53-9
TEST 4 / HIGH ACTIVITY WRITE/READ (128 BYTE/BLOCK MODE)

SEQ 0106

6990	022732	162765	004000	000064
6991	022740	066565	000066	000064
6992	022746	006265	000066	
6993	022752	103402		
6994	022754	000137	021456	
6995	022760	005237	003324	
6996	022764	000207		
6997	022766	000000		
6998	022770	125252		
6999	022772	177777		
7000	022774	052525		
7001	022776	000000		
7002				
7003				
7004	023000			
(3)	023000			
(3)	023000	104401		

```

3: SUB #4000,REC(R5) ;RESTORE TO NEXT RECORD
   ADD TMP(R5),REC(R5) ;HALF INTO REST OF TAPE
   ASR TMP(R5) ;HALF OF HALF FOR NEXT TIME
   BCS 4: ;DONE?
   JMP 1: ;NO
4: INC DONE ;YES-SET FLAG
   RETURN
TST3PT: .WORD 000000
        .WORD 125252
        .WORD 177777
        .WORD 052525
        .WORD 000000

```

ENDTST

L10017: TRAP C\$ETST

```

7007
7008
7009
7010 023002
(3) 023002
7011 023002
(1) 023002 012737 023046 003330
(1) 023010 004737 006024
(1) 023014 004737 005652
(1) 023020 004737 006072
(1) 023024 004737 005532
(1) 023030 103004
(1) 023032 004737 006024
(1) 023036 004737 006072
(1) 023042
7012 023042
(3) 023042 104432
(3) 023044 000724
7013
7014
7015 023046 005065 000064
7016 023052 013765 003312 000066
7017 023060 005065 000062
7018 023064 016565 000064 000072
7019 023072 005737 002220
7020 023076 001403
7021 023100 066565 000060 000072
7022 023106
(1) 023106 012700 027746
(1) 023112 112710 000002
(1) 023116 112760 000012 000001
(1) 023124 112760 000003 000002
(1) 023132 112760 000000 000003
(1) 023140 116560 000060 000004
(1) 023146 112760 000020 000005
(1) 023154 005060 000006
(1) 023160 012760 001000 000010
(1) 023166 016560 000064 000012
(1) 023174 012701 000012
(1) 023200 005721
(1) 023202 012765 000016 000070
(1) 023210 004737 013674
(1) 023214 010110
(1)
(1) 023216 012765 000020 000034
(1) 023224 012765 000001 000036
(1) 023232 012765 000001 000032
(1) 023240 012702 001000
(1) 023244 004737 006600
(1) 023250 032715 000010
(1) 023254 001314
(1) 023256 042715 010000
(1) 023262 012700 027746
(1) 023266 020227 000200
(1) 023272 101004
(1) 023274 010201

.SBTYL TEST 5 / WRITE SELECTED NUMBER OF BLOCKS
BSNTST
TSTID @YST5
MOV @TST5,TSTTOP ;SAVE ADDR OF TEST
CALL SETUP ;INIT UNITS TSTPC
CALL SETDR ;GET 1ST DRVS.
CALL RUN ;DO TEST
CALL SWAPDR ;GET NEXT DRVS.
BCC 64$ ;BR NO 2ND DRVS
CALL SETUP ;REINIT UNITS TSTPC
CALL RUN ;REPEAT TEST
;DONE
64$:
EXIT TST
TRAP C$EXIT
.WORD L10020-.

YST5: CLR REC(R5) ;START AT REC 0
MOV TAPLEN,TMP(R5) ;GET THE # OF BLOCKS PER TRACK
CLR TRK(R5) ;TRK(R5)=1ST OR 2ND PASS COUNTER
1$: MOV REC(R5),PATTEN(R5) ;USE RECORD NO. FOR DATA
TST DRVCHK ;ADD DR #?
BEQ 10$ ;NO
ADD DR(R5),PATTEN(R5) ;YES, ADD DRIVE ID
10$: TUVWIT PATTEN(R5),REC(R5),#512,DR(R5),#0
72$: MOV @TRBUF,P0 ;MAKE COMMAND PACKET:
MOVB @RSCHND,@R0 ;COMMAND FLAG
MOVB @RSMSIZ,1(R0) ;THIS SIZE
MOVB @RSSWR,2(R0) ;INSERT OP CODE-WRITE
MOVB #0,3(R0) ;VERIFY (1 OR 0)
MOVB DR(R5),4(R0) ;DRIVE #
MOVB #020,5(R0) ;MAINTENANCE MODE SWITCH
CLR 6(R0) ;NO SEQUENCE #
MOV #512,8(R0) ;TOTAL COUNT TO WRITE
MOV REC(R5),10(R0) ;AT RECORD N
MOV @RSMSIZ,R1 ;THE PACKET SIZE PLUS 2
TST (R1)+ ;(FLAG AND COUNT) INTO R
MOV @RSMSZ,SNOCNT(R5) ;LOAD THE SIZE TO S
CALL CHKSUM ;R0 --> R1-COUNT
MOV R1,(R0) ;PUT CHKSUM IN PACKET
;SET UP EXPECTATIONS:
MOV @RSCONT,XSFLG(R5) ;THE FLAG
MOV #1,XSCNT(R5) ;THE COUNT
MOV #1,XSPKMN(R5) ;THE # PACKETS EXPECTED
MOV #512,R2 ;GET # OF DATA B
CALL RSVP ;SEND (AND RETURN TO SCH
BIT @BIT3,@R5 ;FLAG BYTE ERROR?
BNE 72$ ;YES
BIC @BIT12,@R5 ;FLAG FOR LAST PACKET
64$: MOV @TRBUF,R0 ;POINT TO TOP OF BUFFER
CMP R2,#128. ;START DATA PACKET(S)
BHI 65$ ;#512. > 128.!
MOV R2,R1 ;#512.<128.

```

MISCELLANEOUS SECTIONS MACY11 30(1046) 25-JAN-84 08:33 PAGE 50 1
CZTUUF.P11 25-JAN-84 08:09

TEST 5 / WRITE SELECTED NUMBER OF BLOCKS

SEQ 0108

(1)	023276	052715	010000			BIS	#BIT12,R5	;SO LAST PACKET NOW
(1)	023302	000402				BR	66#	;USE REMAINING COUNT
(1)	023304	012701	000200		65#:	MOV	#128.,R1	;USE 128. BYTES
(1)	023310	110160	000001		66#:	MOVB	R1,1(R0)	;COPY COUNT TO BUFFER
(1)	023314	010103				MOV	R1,R3	;R3=COUNTER TO LOAD BUFF
(1)	023316	112710	000001			MOVB	#RSDATA,R0	;FLAG FIRST
(1)	023322	005720				TST	(R0)+	;SKIP COUNT
(1)	023324	116520	000072		67#:	MCVB	PATTEN(R5),(R0)+	;INSERT DATA
(1)	023330	005303				DEC	R3	;MORE?
(1)	023332	101374				BHI	67#	;YES
(1)	023334	012700	027746			MOV	#TRBUF,R0	;-->TOP AGAIN
(1)	023340	116001	000001			MOVB	1(R0),R1	;GET COUNT
(1)	023344	042701	177400			BIC	#177400,R1	;ZERO SIGN EXTEND
(1)	023350	010165	000070			MOV	R1,SND CNT(R5)	;HOW MANY TO SEND PLUS
(1)	023354	062765	000004	000070		ADD	#4,SND CNT(R5)	;FLAG,COUNT,CHKSUM
(1)	023362	062701	000002			ADD	#2,R1	;COMPENSATE FOR FLAG + C
(1)	023366	004737	013674			CALL	CHKSUM	;FOR CHECKSUM CALC.
(1)	023372	110120				MOVB	R1,(R0)+	;CHKSUM INTO PACKET
(1)	023374	000301				SWAB	R1	;EVEN ON AN ODD
(1)	023376	110120				MOVB	R1,(R0)+	;BYTE BOUNDARY
(1)	023400	032715	010000			BIT	#BIT12,R5	;LAST DATA PACKET?
(1)	023404	001412				BEG	68#	;NO
(1)	023406	012765	000002	000034		MOV	#RSEND,XSFLG(R5)	;YES-EXPECT 'END'
(1)	023414	012765	000016	000036		MOV	#RSND SZ,XSCNT(R5)	;OF THIS SIZE
(1)	023422	012765	000001	000032		MOV	#1,XSPK M(R5)	;AND 1 PACKET
(1)	023430	000411				BR	69#	;SEND
(1)	023432	012765	000020	000034	68#:	MOV	#RSCONT,XSFLG(R5)	;(NOT LAST). EXPECT
(1)	023440	012765	000001	000036		MOV	#1,XSCNT(R5)	;AND 1 BYTE
(1)	023446	012765	000001	000032		MOV	#1,XSPK M(R5)	;AND 1 PACKET
(1)	023454	004737	006600		69#:	CALL	RSVP	;SEND PACKET
(1)								;AND RETURN TO SCHEDULER
(1)	023460	032715	000010			BIT	#BIT3,R5	;FLAG BYTE RETRY?
(1)	023464	001210				BNE	72#	;YES
(1)	023466	032715	002000			BIT	#BIT10,R5	;RETRY DATA ERROR?
(1)	023472	001004				BNE	70#	;YES
(1)	023474	162702	000200			SUB	#128.,R2	;NO, MORE DATA TO SEND?
(1)	023500	101270				BHI	64#	;YES
(1)	023502	000502				BR	71#	;NO
(1)	023504				70#:	TURTRY	REC(R5),#512.,DR(R5)	;RETRY HERE
(2)								
(2)								
(2)	023504	012700	027746		76#:	MOV	#TRBUF,R0	;FORM CMND PACK;
(2)	023510	112710	000002			MOVB	#PSCMND,R0	;MESSAGE PACK TYPE
(2)	023514	112760	000012	000001		MOVB	#RMSIZ,1(R0)	;THIS BIG
(2)	023522	112760	000002	000002		MOVB	#RSSRD,2(R0)	;OP CODE-READ
(2)	023530	016560	000064	000012		MOV	REC(R5),10.(R0)	;THIS RECORD
(2)	023536	116560	000060	000004		MOVB	DR(R5),4.(R0)	;THIS DRIVE
(2)	023544	105060	000003			CLRB	3(R0)	;PRESET NORM THRESHOLD
(2)	023550	105715				TSTB	R5	;REDUCED?
(2)	023552	100002				BPL	77#	;NO
(2)	023554	105260	000003			INCB	3(R0)	;YES-CHANGE THRESHOLD
(2)	023560	012760	001000	000010	77#:	MOV	#512.,8.(R0)	;# BYTES DESIRED
(2)	023566	112760	000020	000005		MOVB	#020,5.(R0)	;MAINTENANCE MODE
(2)	023574	005060	000006			CLR	6.(R0)	;NO SEQUENCE #
(2)	023600	012701	000012			MOV	#RMSIZ,R1	;SIZE OF PACKET
(2)	023604	005721				TST	(R1)+	;PLUS FLAG+COUNT INTO R1

MISCELLANEOUS SECTIONS MACY11 30(1046) 25-JAN-84 08:09
CZTUUF.P11 25-JAN-84 08:09

25-JAN-84 08:33 PAGE 54-2
TEST 5 / WRITE SELECTED NUMBER OF BLOCKS

SEQ 0109

(2)	023606	012765	000016	000070		MOV	#RSSNSZ,SNDCNT(R5)	;SET UP SIZE TO SEND
(2)	023614	004737	013674			CALL	CHKSUM	;FORM CHECKSUM R1=COUNT
(2)	023620	010110				MOV	R1,(R0)	;INSERT IN PACKET
(2)	023622	012701	001000			MOV	#512.,R1	;SET EXPECTATION
(2)	023626	012703	000034			MOV	#XSFLG,R3	;CALC # OF DATA PACKETS
(2)	023632	060503				ADD	R5,R3	;OFFSET OF FLAG
(2)	023634	005002				CLR	R2	;ABS. ADDR. OF XSFLG
(2)	023636	005202				CLR	R2	;PRESET
(2)	023640	012723	000001		73\$:	INC	R2	;# PACKETS EXPECTED
(2)	023644	012723	000204			MOV	#RSDATA,(R3)+	;LOAD XSFLG
(2)	023650	162701	000200			MOV	#132.,(R3)+	;AND EXPECT COUNT
(2)	023654	101401				SUB	#128.,R1	;NEG RESULT LAST TIME
(2)	023656	000757				BLOS	75\$;LAST TIME!
(2)	023660	005202				BR	73\$;MORE TO DO
(2)	023662	010265	000032		75\$:	INC	R2	;ADD ONE FOR END PACK
(2)	023666	012723	000002			MOV	R2,XSPKMM(R5)	;SAVE # PACKETS TO EXPECT
(2)	023672	012713	000016			MOV	#RSEND,(R3)+	;EXPECT AN END
(2)	023676	004737	006600			MOV	#RSNDSZ,(R3)	;THIS BIG-14. BYTES
(2)						CALL	RSVP	;SEND
(2)								;AND RETURN TO SCHEDULER
7023	023712	005365	000066			DEC	TMP(R5)	;DO ALL RECORDS FOR THIS TRACK?
7024	023716	001404				BEQ	2\$;YES-GET OTHER TRACK
7025	023720	005265	000064			INC	REC(R5)	;NO-ONTO NEXT RECORD
7026	023724	000137	023064			JMP	1\$;EXECUTE THE WRITE
7027	023730	005765	000062		2\$:	TST	TRK(R5)	;DONE 2 TRACKS?
7028	023734	001012				BNE	TSTSEX	;YES-EXIT
7029	023736	005255	000062			INC	TRK(R5)	;NO-SET FLAG FOR NEXT PASS
7030	023742	013765	003336	000064		MOV	SECREC,REC(R5)	;GET NEW STARTING BLOCK #
7031	023750	013765	003312	000066		MOV	TAPLEN,TMP(R5)	;RESET # OF BLOCKS
7032	023756	000137	023064			JMP	1\$;AND EXECUTE
7033	023762	005237	003324		TSTSEX:	INC	DONE	;DONE
7034	023766	000207				RETURN		;RETURN
7035								
7036	023770					ENDTST		
(3)	023770							
(3)	023770	104401						

L10020: TRAP C\$ETST

```

7039          .SBTTL TEST 6 / READ SELECTED NUMBER OF BLOCKS
7040
7041 023772          BGNTST
(3) 023772
7042 023772          TSTID  #TST6
(1) 023772 012737 024036 003330          MOV  #T6,TSTTOP      ;SAVE ADDR OF TEST
(1) 024000 004737 006024          CALL  SETUP          ;INIT UNITS TSTPC
(1) 024004 004737 005652          CALL  SETDR          ;GET 1ST DRVS.
(1) 024010 004737 006072          CALL  RUN            ;DO TEST
(1) 024014 004737 005532          CALL  SWAPDR         ;GET NEXT DRVS.
(1) 024020 103004          BCC  64$            ;BR NO 2ND DRVS
(1) 024022 004737 006024          CALL  SETUP          ;REINIT UNITS TSTPC
(1) 024026 004737 006072          CALL  RUN            ;REPEAT TEST
(1) 024032          64$:          ;DONE
7043 024032          EXIT TST
(3) 024032 104432          TRAP  C$EXIT
(3) 024034 000520          .WORD L10021-.
7044
7045
7046 024036 005065 000064          TST6: CLR  REC(R5)      ;START AT REC 0
7047 024042 013765 003312 000066          MOV  TAPLEN,TMP(R5) ;GET THE # OF BLOCKS PER TRACK
7048 024050 005065 000062          CLR  TRK(R5)        ;TRK(R5)=1ST OR 2ND PASS
7049 024054 016565 000064 000072          1$:  MOV  REC(R5),PATTEN(R5) ;USE RECORD NO. AS DATA
7050 024062 005737 002220          TST  DRVCHK          ;ADD DR #?
7051 024066 001403          BEQ  10$            ;NO
7052 024070 066565 000060 000072          ADD  DR(R5),PATTEN(R5) ;ADD IN DRIVE ID
7053 024076          10$:  TUREAD REC(R5),#512.,DR(R5),#0
(1)
(1)
(1) 024076 012700 027746          68$:  MOV  #TRBUF,R0      ;FORM CMND PACK:
(1) 024102 112710 000002          MOVB #RSCMND,R0     ;MESSAGE PACK TYPE
(1) 024106 112760 000012 000001          MOVB #RSMISZ,1(R0) ;THIS BIG
(1) 024114 112760 000002 000002          MOVB #RSSRD,2(R0)  ;OP CODE IS READ
(1) 024122 016560 000064 000012          MOV  REC(R5),10.(R0) ;THIS RECORD
(1) 024130 116560 000060 000004          MOVB DR(R5),4.(R0)  ;THIS DRIVE
(1) 024136 112760 000000 000003          MOVB #0,3.(R0)     ;VERIFY
(1) 024144 012760 001000 000010          MOV  #512.,8.(R0)  ;TOTAL BYTES TO READ
(1) 024152 112760 000020 000005          MOVB #020,5.(R0)   ;MAINTENANCE MODE
(1) 024160 005060 000006          CLR  6.(R0)         ;NO SEQUENCE #
(1) 024164 012701 000012          MOV  #RSMISZ,R1     ;GET SIZE OF PACKET
(1) 024170 005721          TST  (R1)+          ;+2 FOR CHECKSUM
(1) 024172 012765 000016 000070          MOV  #RSSNSZ,SNDCNT(R5) ;SIZE TO SEND
(1) 024200 004737 013674          CALL  CHKSUM        ;FORM CHECKSUM R1+COUNT
(1) 024204 010110          MOV  R1,(R0)        ;INSERT CHECKSUM
(1)
(1) 024206 012701 001000          MOV  #512.,R1      ;SET EXPECTATION
(1)
(1) 024212 012703 000034          MOV  #XSFLG,R3     ;CALC # OF DATA PACKETS
(1) 024216 060503          ADD  R5,R3         ;GET OFFSET
(1) 024220 005002          CLR  R2            ;ABS. ADDR. OF XSFLG
(1) 024222 005202          INC  R2            ;PRESET AS NONE
(1) 024224 012723 000001          64$:  INC  R2            ;# PACKETS EXPECTED
(1) 024230 012723 000204          MOV  #RSDATA,(R3)+ ;LOAD XSFLG
(1) 024234 162701 000200          MOV  #132.,(R3)+  ;ADD EXPECTED COUNT
(1) 024240 101401          SUB  #128.,R1      ;NEG RESULT LAST TIME
(1) 024242 000767          BLOS 66$           ;LAST TIME
          BR  64$           ;MORE TO DO

```

(1)	024244	005202				66#:	INC R2	;ADD ONE FOR END PACK
(1)	024246	010265	000032				MOV R2,XSPKNM(R5)	;SAVE # PACKETS TO EXPECT
(1)	024252	012723	000002				MOV #RSEND,(R3)+	;EXPECT AN END ALSO...
(1)	024256	C12713	000016				MOJ #RSNDSZ,(R3)	;THIS BIG-14. BYTES
(1)	024262	004737	006600				CALL RSVP	;SEND
(1)								;AND RETURN TO SCHEDULER
(1)	024266	032715	002010			67#:	BIT #BIT10:BIT3,#R5	;RETRY?
(1)	024272	001500					BEQ 65#	;NO.
(1)	024274						TURTRY REC(R5),#512.,DR(R5)	;YES
(2)								
(2)								
(2)	024274	012700	027746			72#:	MOV #TRBUF,R0	;FORM CMND PACK:
(2)	024300	112710	000002				MOV #RSCMND,#R0	;MESSAGE PACK TYPE
(2)	024304	112760	000012	000001			MOV #RSMSIZ,1(R0)	;THIS BIG
(2)	024312	112760	000002	000002			MOV #RSSRD,2(R0)	;OP CODE-READ
(2)	024320	016560	000064	000012			MOV REC(R5),10.(R0)	;THIS RECORD
(2)	024326	116560	000060	000004			MOV DR(R5),4.(R0)	;THIS DRIVE
(2)	024334	105060	000003				CLRB 3(R0)	;PRESET NORM THRESHOLD
(2)	024340	105715					TSTB #R5	;REDUCED?
(2)	024342	100002					BPL 73#	;NO
(2)	024344	105260	000003				INCB 3(R0)	;YES-CHANGE THRESHOLD
(2)	024350	012760	001000	000010		73#:	MOV #512.,8.(R0)	;# BYTES DESIRED
(2)	024356	112760	000020	000005			MOV #020,5.(R0)	;MAINTENANCE MODE
(2)	024364	005060	000006				CLR 6.(R0)	;NO SEQUENCE #
(2)	024370	012701	000012				MOV #RSMSIZ,R1	;SIZE OF PACKET
(2)	024374	005721					TST (R1)+	;PLUS FLAG+COUNT INTO R1
(2)	024376	012765	000016	000070			MOV #RSSNSZ,SND CNT(R5)	;SET UP SIZE TO SEND
(2)								
(2)	024404	004737	013674				CALL CHKSUM	;FORM CHECKSUM R1-COUNT
(2)	024410	010110					MOV R1,(R0)	;INSERT IN PACKET
(2)								
(2)	024412	012701	001000				MOV #512.,R1	;SET EXPECTATION
(2)								;CALC # OF DATA PACKETS
(2)	024416	012703	000034				MOV #XSFLG,R3	;OFFSET OF FLAG
(2)	024422	060503					ADD R5,R3	;ABS. ADDR. OF XSFLG
(2)	024424	005002					CLR R2	;PRESET
(2)	024426	005202					INC R2	;# PACKETS EXPECTED
(2)	024430	012723	000001			69#:	MOV #RSDATA,(R3)+	;LOAD XSFLG
(2)	024434	012723	000204				MOV #132.,(R3)+	;AND EXPECT COUNT
(2)	024440	162701	000200				SUB #128.,R1	;NEG RESULT LAST TIME
(2)	024444	101401					BLOS 71#	;LAST TIME!
(2)	024446	000767					BR 69#	;MORE TO DO
(2)	024450	005202						;ADD ONE FOR END PACK
(2)	024452	010265	000032			71#:	INC R2	;SAVE # PACKETS TO EXPECT
(2)	024456	012723	000002				MOV #RSEND,(R3)+	;EXPECT AN END
(2)	024462	012713	000016				MOV #RSNDSZ,(R3)	;THIS BIG-14. BYTES
(2)								
(2)	024466	004737	006600				CALL RSVP	;SEND
(2)								;AND RETURN TO SCHEDULER
(2)								
7054	024476	005365	000066				DEC IMP(R5)	;DO ALL RECORDS THIS TRACK?
7055	024502	001404					BEQ 2#	;YES-GET OTHER TRACK
7056	024504	005265	000064				INC REC(R5)	;NO-NEXT RECORD
7057	024510	000137	024054				JMP 1#	;EXECUTE THE READ
7058	024514	005765	000062			2#:	TST TRK(R5)	;DONE 2 TRACKS?
7059	024520	001012					BNE TST6EX	;YES-EXIT

MISCELLANEOUS SECTIONS MACY11 30(1046) 25-JAN-84 08:33 PAGE 55-2
CZTUUF.P11 25-JAN-84 08:09 TEST 6 / READ SELECTED NUMBER OF BLOCKS

SEQ 0112

7060	024522	005265	000062		INC	TRK(R5)	;NO-SET FLAG FOR NEXT PASS
7061	024526	013765	003336	000064	MOV	SECRC,REC(R5)	;GET NEW STARTING BLOCK #
7062	024534	013765	003312	000066	MOV	TAPLEN,TMP(R5)	;RESET # OF BLOCKS
7063	024542	000137	024054		JMP	1\$;AND EXECUTE
7064	024546	005237	003324		TEST6EX: INC	DONE	;DONE
7065	024552	000207			RETURN		;RETURN
7066							
7067	024554				ENDTST		
(3)	024554						
(3)	024554	104401					

L10021: TRAP CGETST


```

7070          .SBTTL TEST 7 / WRITE-VERIFY SELECTED NUMBER OF BLOCKS
7071
7072 024556          BGNTST
(3) 024556
7073 024556          TSTID  #TST7
(1) 024556 012737 024622 003330          MOV  #TST7,TSTTOP      ;SAVE ADDR OF TEST
(1) 024564 004737 006024          CALL  SETUP          ;INIT UNITS TSTPC
(1) 024570 004737 005652          CALL  SETDR         ;GET 1ST DRVS.
(1) 024574 004737 006072          CALL  RUN           ;DO TEST
(1) 024600 004737 005532          CALL  SWAPDR        ;GET NEXT DRVS.
(1) 024604 103004          BCC   64$          ;BR NO 2ND DRVS
(1) 024606 004737 006024          CALL  SETUP         ;REINIT UNITS TSTPC
(1) 024612 004737 006072          CALL  RUN           ;REPEAT TEST
(1) 024616          64$:          ;DONE
7074 024616          EXIT TST
(3) 024616 104432          TRAP  C$EXIT
(3) 024620 000724          .WORD L10022-.
7075
7076
7077 024622 005065 000064          TST7: CLR  REC(R5)      ;START AT REC 0
7078 024626 013765 003312 000066          MOV  TAPLEN,TMP(R5) ;GET THE # OF BLOCKS PER TRACK
7079 024634 005065 000062          CLR  TRK(R5)        ;TRK(R5)=1ST OR 2ND PASS
7080 024640 016565 000064 000072          1$:  MOV  REC(R5),PATTEN(R5) ;USE RECORD NO. FOR DATA
7081 024646 005737 002220          TST  DRVCHK         ;ADD DR #?
7082 024652 001403          BEQ  10$          ;NO
7083 024654 066565 000060 000072          ADD  DR(R5),PATTEN(R5) ;ADD DRIVE ID
7084 024662          10$:  TUIWRIT PATTEN(R5),REC(R5),#512.,DR(R5),#1
(1) 024662 012700 027746          72$:  MOV  #TRBUF,RO      ;MAKE COMMAND PACKET:
(1) 024666 112710 000002          MOV  #RSCMD,RO      ;COMMAND FLAG
(1) 024672 112760 000012 000001          MOV  #RSMSIZ,1(RO) ;THIS SIZE
(1) 024700 112760 000003 000002          MOV  #RSSWR,2(RO)  ;INSERT OP CODE-WRITE
(1) 024706 112760 000001 000003          MOV  #1,3.(RO)     ;VERIFY (1 OR 0)
(1) 024714 116560 000060 000004          MOV  DR(R5),4.(RO) ;DRIVE #
(1) 024722 112760 000020 000005          MOV  #020.5.(RO)  ;MAINTENANCE MODE SWITCH
(1) 024730 005060 000006          CLR  6.(RO)        ;NO SEQUENCE #
(1) 024734 012760 001000 000010          MOV  #512.,8.(RO) ;TOTAL COUNT TO WRITE
(1) 024742 016560 000064 000012          MOV  REC(R5),10.(RO) ;AT RECORD N
(1) 024750 012701 000012          MOV  #RSMSIZ,R1    ;THE PACKET SIZE PLUS+2
(1) 024754 005721          TST  (R1)+         ;(FLAG AND COUNT) INTO R
(1) 024756 012765 000016 000070          MOV  #RSSNSZ,SNDCNT(R5) ;LOAD THE SIZE TO S
(1) 024764 004737 013674          CALL CHKSUM        ;R0 --> R1=COUNT
(1) 024770 010110          MOV  R1,(RO)       ;PUT CHKSUM IN PACKET
(1)          ;SET UP EXPECTATIONS:
(1) 024772 012765 000020 000034          MOV  #RSCONT,XSFLG(R5) ;THE FLAG
(1) 025000 012765 000001 000036          MOV  #1,XSCNT(R5)   ;THE COUNT
(1) 025006 012765 000001 000032          MOV  #1,XSPKNM(R5) ;THE # PACKETS EXPECTED
(1) 025014 012702 001000          MOV  #512.,R2      ;GET # OF DATA B
(1) 025020 004737 006600          CALL  RSVP         ;SEND (AND RETURN TO SCH
(1) 025024 032715 000010          BIT  #BIT3,DR5     ;FLAG BYTE ERROR?
(1) 025030 001314          BNE  72$          ;YES
(1) 025032 042715 010000          BIC  #BIT12,DR5    ;FLAG FOR LAST PACKET
(1) 025036 012700 027746          64$:  MOV  #TRBUF,RO      ;POINT TO TOP OF BUFFER
(1) 025042 020227 000200          CMP  R2,#128.     ;START DATA PACKET(S)
(1) 025046 101004          BHI  65$          ;#512. > 128.!
(1) 025050 010201          MOV  R2,R1        ;#512.<128.
(1) 025052 052715 010000          BIS  #BIT12,DR5    ;SO LAST PACKET NOW

```

Line	Address	Offset	Count	Label	Instruction	Comment
(1)	025056	000402			BR	66\$;USE REMAINING COUNT
(1)	025060	012701	000200	65\$:	MOV	0128.,R1 ;USE 128. BYTES
(1)	025064	110160	000001	66\$:	MOVB	R1,1(R0) ;COPY COUNT TO BUFFER
(1)	025070	010103			MOV	R1,R3 ;R3=COUNTER TO LOAD BUFF
(1)	025072	112710	000001		MOVB	0RSDATA,0R0 ;FLAG FIRST
(1)	025076	005720			TST	(R0)+ ;SKIP COUNT
(1)	025100	116520	000072	67\$:	MOVB	PATTEN(R5),(R0)+ ;INSERT DATA
(1)	025104	005303			DEC	R3 ;MORE?
(1)	025106	101374			BHI	67\$;YES
(1)	025110	012700	027746		MOV	0TRBUF,R0 ;-->TOP AGAIN
(1)	025114	116001	000001		MOVB	1(R0),R1 ;GET COUNT
(1)	025120	042701	177400		BIC	0177400,R1 ;ZERO SIGN EXTEND
(1)	025124	010165	000070		MOV	R1,SNDCNT(R5) ;HOW MANY TO SEND PLUS
(1)	025130	012765	000004 000070		ADD	04,SNDCNT(R5) ;FLAG,COUNT,CHKSUM
(1)	025136	062701	000002		ADD	02,R1 ;COMPENSATE FOR FLAG + C
(1)	025142	004737	013674		CALL	CHKSUM ;FOR CHECKSUM CALC.
(1)	025146	110120			MOVB	R1,(R0)+ ;CHKSUM INTO PACKET
(1)	025150	000301			SWAB	R1 ;EVEN ON AN ODD
(1)	025152	110120			MOVB	R1,(R0)+ ;BYTE BOUNDARY
(1)	025154	032715	010000		BIT	0BIT12,0R5 ;LAST DATA PACKET?
(1)	025160	001412			BEQ	68\$;NO
(1)	025162	012765	000002 000034		MOV	0RSEND,XSFLG(R5) ;YES-EXPECT 'END'
(1)	025170	012765	000016 000036		MOV	0RSNDSZ,XSCNT(R5) ;OF THIS SIZE
(1)	025176	012765	000001 000032		MOV	01,XSPKMN(R5) ;AND 1 PACKET
(1)	025204	000411			BR	69\$;SEND
(1)	025206	012765	000020 000034	68\$:	MOV	0RSCONT,XSFLG(R5) ;(NOT LAST), EXPECT
(1)	025214	012765	000001 000036		MOV	01,XSCNT(R5) ;AND 1 BYTE
(1)	025222	012765	000001 000032		MOV	01,XSPKMN(R5) ;AND 1 PACKET
(1)	025230	004737	006600	69\$:	CALL	RSVP ;SEND PACKET
(1)	025234	032715	000010			;AND RETURN TO SCHEDULER
(1)	025240	001210			BIT	0BIT3,0R5 ;FLAG BY RETRY?
(1)	025242	032715	002000		BNE	72\$;YES
(1)	025246	001004			BIT	0BIT10,0R5 ;RETRY DATA ERROR?
(1)	025250	162702	000200		BNE	70\$;YES
(1)	025254	101270			SUB	0128.,R2 ;NO, MORE DATA TO SEND?
(1)	025256	000502			BHI	64\$;YES
(1)	025260			70\$:	BR	71\$;NO
(2)					TUNTRY	REC(R5),0512.,DR(R5) ;RETRY HERE
(2)						
(2)	025260	012700	027746	76\$:	MOV	0TRBUF,R0 ;FORM CMD PACK:
(2)	025264	112710	000002		MOVB	0RSCMD,0R0 ;MESSAGE PACK TYPE
(2)	025270	112760	000012 000001		MOVB	0RSMIZ,1(R0) ;THIS BIG
(2)	025276	112760	000002 000002		MOVB	0RSSRD,2(R0) ;OP CODE-READ
(2)	025304	016560	000064 000012		MOV	FEC(R5),10.(R0) ;THIS RECORD
(2)	025312	116560	000060 000004		MOVB	DR(R5),4.(R0) ;THIS DRIVE
(2)	025320	105060	000003		CLRB	3(R0) ;PRESET NORM THRESHOLD
(2)	025324	105715			TSTB	0R5 ;REDUCED?
(2)	025326	100002			BPL	77\$;NO
(2)	025330	105260	000003		INCB	3(R0) ;YES-CHANGE THRESHOLD
(2)	025334	012760	001000 000010	77\$:	MOV	0512.,8.(R0) ;# BYTES DESIRED
(2)	025342	112760	000020 000005		MOVB	0020,5.(R0) ;MAINTENANCE MODE
(2)	025350	005060	000006		CLR	6.(R0) ;NO SEQUENCE #
(2)	025354	012701	000012		MOV	0RSMISZ,R1 ;SIZE OF PACKET
(2)	025360	005721			TST	(R1)+ ;PLUS FLAG+COUNT INTO R1
(2)	025362	012765	000016 000070		MOV	0RSSNSZ,SNDCNT(R5) ;SET UP SIZE TO SEND

```

(2)
(2) 025370 004737 013674          CALL   CHKSUM      ;FORM CHECKSUM R1=COUNT
(2) 025374 010110                MOV    R1,(R0)    ;INSERT IN PACKET
(2)
(2) 025376 012701 001000          MOV    #512.,R1   ;SET EXPECTATION
(2)                                ;CALC # OF DATA PACKETS
(2) 025402 012703 000034          MOV    #XSFLG,R3  ;OFFSET OF FLAG
(2) 025406 060503                ADD    R5,R3      ;ABS. ADDR. OF XSFLG
(2) 025410 005002                CLR    R2         ;PRESET
(2) 025412 005202                73$: INC    R2     ;# PACKETS EXPECTED
(2) 025414 012723 000001          MOV    #RSDATA,(R3)+ ;LOAD XSFLG
(2) 025420 012723 000204          MOV    #132.,(R3)+ ;AND EXPECT COUNT
(2) 025424 162701 000200          SUB    #128.,R1   ;NEG RESULT LAST TIME
(2) 025430 101401                BLOS  75$        ;LAST TIME!
(2) 025432 000767                BR    73$        ;MORE TO DO
(2) 025434 005202                75$: INC    R2     ;ADD ONE FOR END PACK
(2) 025436 010265 000032          MOV    R2,XSPKMM(R5) ;SAVE # PACKETS TO EXPECT
(2) 025442 012723 000002          MOV    #RSEND,(R3)+ ;EXPECT AN END
(2) 025446 012713 000016          MOV    #RSNDSZ,(R3) ;THIS BIG-14. BYTES
(2)
(2) 025452 004737 006600          CALL   RSVP      ;SEND
(2)                                ;AND RETURN TO SCHEDULER
(2)
7085 025466 005365 000066          DEC    TMP(R5)   ;DO ALL RECORDS FOR THIS TRACK?
7086 025472 011404                BEQ    2$        ;YES-GET OTHER TRACK
7087 025474 005265 000064          INC    REC(R5)  ;NO-NEXT RECORD
7088 025500 000137 024640          JMP    1$        ;EXECUTE THE WRITE
7089 025504 005765 000062          2$: TST   TRK(R5) ;DONE 2 TRACKS?
7090 025510 001012                BNE    TST7EX   ;YES-EXIT
7091 025512 005265 000062          INC    TRK(R5)  ;NO-SET FLAG FOR NEXT PASS
7092 025516 013765 003336 000064  MOV    SECREC,REC(R5) ;GET NEW STARTING BLOCK #
7093 025524 013765 003312 000066  MOV    TAPLEN,TMP(R5) ;RESET # OF BLOCKS
7094 025532 000137 024640          JMP    1$        ;AND EXECUTE
7095 025536 005237 003324          TST7EX: INC    DONE ;DONE
7096 025542 000207                RETURN          ;RETURN
7097
7098 025544                ENDTST
(3) 025544
(5) 025544 104401                L10022: TRAP    C$ETST

```

```

7101          .SBTTL TEST 8 / READ-REDUCED THRESHOLD SELECTED NUMBER OF BLOCKS
7102
7103 025546    BGNTST
(3) 025546
7104 025546    TSTJD  #TST8
(1) 025546 012737 025612 003330    MOV  #TST8,TSTTOP ;SAVE ADDR OF TEST
(1) 025554 004737 006024    CALL  SETUP      ;INIT UNITS TSTPC
(1) 025560 004737 005652    CALL  SETDR      ;GET 1ST DRVS.
(1) 025564 004737 006072    CALL  RUN        ;DO TEST
(1) 025570 004737 005532    CALL  SWAPDR     ;GET NEXT DRVS.
(1) 025574 103004    BCC  64$        ;BR NO 2ND DRVS
(1) 025576 004737 006024    CALL  SETUP      ;REINIT UNITS TSTPC
(1) 025602 004737 006072    CALL  RUN        ;REPEAT TEST
(1) 025606    64$:        ;DONE
7105 025606    EXIT TST
(3) 025606 104432    TRAP  C$EXIT
(3) 025610 000520    .WORD L10023-.
7106
7107
7108 025612 005065 000064    TST8: CLR  REC(R5) ;START AT REC 0
7109 025616 013765 003312 000066    MOV  TAPLEN,MP(R5) ;GET THE # OF BLOCKS PER TRACK
7110 025624 005065 000062    CLR  TRK(R5) ;TRK(R5)=1ST OR 2ND PASS
7111 025630 016565 000064 000072    1$:  MOV  REC(R5),PATTEN(R5) ;USE RECORD NO. FOR DATA
7112 025636 005737 002220    TST  DRVCHK ;ADD DR #?
7113 025642 001403    BEQ  10$        ;NO
7114 025644 066565 000060 000072    ADD  DR(R5),PATTEN(R5) ;ADD DRIVE ID
7115 025652    10$:  TUREAD REC(R5),#512.,DR(R5),#1
(1)
(1)
(1) 025652 012700 027746    68$:  MOV  #TRBUF,R0 ;FORM CMND PACK:
(1) 025656 112710 000002    MOVB #RSCMND,#R0 ;MESSAGE PACK TYPE
(1) 025662 112760 000012 000001    MOVB #RSMISZ,1(R0) ;THIS BIG
(1) 025670 112760 000002 000002    MOVB #RSSRD,2(R0) ;OP CODE IS READ
(1) 025676 016560 000064 000012    MOV  REC(R5),10.(R0) ;THIS RECORD
(1) 025704 116560 000060 000004    MOVB DR(R5),4.(R0) ;THIS DRIVE
(1) 025712 112760 000003 000003    MOVB #1,3.(R0) ;VERIFY
(1) 025720 012760 001000 000010    MOV  #512.,8.(R0) ;TOTAL BYTES TO READ
(1) 025726 112760 000020 000005    MOVB #020,5.(R0) ;MAINTENANCE MODE
(1) 025734 005060 000006    CLR  6.(R0) ;NO SEQUENCE #
(1) 025740 012701 000012    MOV  #RSMISZ,R1 ;GET SIZE OF PACKET
(1) 025744 005721    TST  (R1)+ ;+2 FOR CHECKSUM
(1) 025746 012765 000016 000070    MOV  #RSSNSZ,SNDCNT(R5) ;SIZE TO SEND
(1) 025754 004737 013674    CALL CHKSUM ;FORM CHECKSUM R1*COUNT
(1) 025760 010110    MOV  R1,(R0) ;INSERT CHECKSUM
(1)
(1) 025762 012701 001000    MOV  #512.,R1 ;SET EXPECTATION
(1)
(1) 025766 012703 000034    ;CALC # OF DATA PACKETS
(1) 025772 060503    MOV  #XSFLG,R3 ;GET OFFSET
(1) 025774 005002    ADD  R5,R3 ;ABS. ADDR. OF XSFLG
(1) 025776 005202    CLR  R2 ;PRESET AS NONE
(1) 026000 012723 000001    64$:  INC  R2 ;# PACKETS EXPECTED
(1) 026004 012723 000204    MOV  #RSDATA,(R3)+ ;LOAD XSFLG
(1) 026010 162701 000200    MOV  #132.,(R3)+ ;AND EXPECTED COUNT
(1) 026014 101401    SUB  #128.,R1 ;NEG RESULT LAST TIME
(1) 026016 000767    BLOS 66$ ;LAST TIME
;MORE TO DO

```

(1)	026020	005202			66\$:	INC R2	;ADD ONE FOR END PACK
(1)	026022	010265	000032			MOV R2,XSPKMN(R5)	;SAVE # PACKETS TO EXPEC
(1)	026026	012723	000002			MOV #RSEND,(R3)+	;EXPECT AN END ALSO...
(1)	026032	012713	000016			MOV #RSNDSZ,(R3)	;THIS BIG-14. BYTES
(1)	026036	004737	006600			CALL RSVP	;SEND
(1)							;AND RETURN TO SCHEDULER
(1)	026042	032715	002010		67\$:	BIT #BIT10!BIT3,#R5	;RETRY?
(1)	026046	001500				BEQ 65\$;NO.
(1)	026050					TURTRY REC(R5),#512.,DR(R5)	;YES
(2)							
(2)							
(2)	026050	012700	027746		72\$:	MOV #TRBUF,R0	;FORM CMND PACK;
(2)	026054	112710	000002			MOVB #RSCMND,#R0	;MESSAGE PACK TYPE
(2)	026060	112760	000012	000001		MOVB #RSMISZ,1(R0)	;THIS BIG
(2)	026066	112760	000002	000002		MOVB #RSSRD,2(R0)	;OP CODE-READ
(2)	026074	016560	000064	000012		MOV REC(R5),10.(R0)	;THIS RECORD
(2)	026102	116560	000060	000004		MOVB DR(R5),4.(R0)	;THIS DRIVE
(2)	026110	105060	000003			CLRB 3(R0)	;PRESET NORM THRESHOLD
(2)	026114	105715				TST #R5	;REDUCED?
(2)	026116	100002				BPL 73\$;NO
(2)	026120	105260	000003			INCB 3(R0)	;YES-CHANGE THRESHOLD
(2)	026124	012760	001000	000010	73\$:	MOV #512.,8.(R0)	;# BYTES DESIRED
(2)	026132	112760	000020	000005		MOVB #020,5.(R0)	;MAINTENANCE MODE
(2)	026140	005060	000006			CLR 6.(R0)	;NO SEQUENCE #
(2)	026144	012701	000012			MOV #RSMISZ,R1	;SIZE OF PACKET
(2)	026150	005721				TST (R1)+	;PLUS FLAG+COUNT INTO R1
(2)	026152	012765	000010	000070		MOV #RSSNSZ,SND CNT(R5)	;SET UP SIZE TO SEND
(2)							
(2)	026160	004737	013674			CALL CHKSUM	;FORM CHECKSUM R1=COUNT
(2)	026164	010110				MOV R1,(R0)	;INSERT IN PACKET
(2)							
(2)	026166	012701	001000			MOV #512.,R1	;SET EXPECTATION
(2)							;CALC # OF DATA PACKETS
(2)	026172	012703	000034			MOV #XSFLG,R3	;OFFSET OF FLAG
(2)	026176	060503				ADD R5,R3	;ABS. ADDR. OF XSFLG
(2)	026200	005002				CLR R2	;PRESET
(2)	026202	005202			69\$:	INC R2	;# PACKETS EXPECTED
(2)	026204	012723	000001			MOV #RSDATA,(R3)+	;LOAD XSFLG
(2)	026210	012723	000204			MOV #132.,(R3)+	;AND EXPECT COUNT
(2)	026214	162701	000200			SUB #126.,R1	;NEG RESULT LAST TIME
(2)	026220	101401				BLOS 71\$;LAST TIME!
(2)	026222	000767				BR 69\$;MORE TO DO
(2)	026224	005202			71\$:	INC R2	;ADD ONE FOR END PACK
(2)	026226	010265	000032			MOV R2,XSPKMN(R5)	;SAVE # PACKETS TO EXPEC
(2)	026232	012723	000002			MOV #RSEND,(R3)+	;EXPECT AN END
(2)	026236	012713	000016			MOV #RSNDSZ,(R3)	;THIS BIG-14. BYTES
(2)							
(2)	026242	004737	006600			CALL RSVP	;SEND
(2)							;AND RETURN TO SCHEDULER
(2)							
7116	026252	005365	000066			DEC TMP(R5)	;DO ALL RECORDS THIS TRACK?
7117	026256	001404				BEQ 2\$;YES-GET OTHER TRACK
7118	026260	005265	000064			INC REC(R5)	;NO-NEXT RECORD
7119	026264	000137	025630			JMP 1\$;EXECUTE THE READ
7120	026270	005765	000062		2\$:	TST TRK(R5)	;DONE 2 TRACKS?
7121	026274	001012				BNE TSTBEX	;YES-EXIT

B10

MISCELLANEOUS SECTIONS MACY11 30(1046)
CZTUUF.P11 25-JAN-84 08:09

25-JAN-84 08:33 PAGE 57-2
TEST 8 / READ-REDUCED THRESHOLD SELECTED NUMBER OF BLOCKS

SEQ 0118

7122	026276	005265	000062	
7123	026302	013765	003336	000064
7124	026310	013765	003312	000066
7125	026316	000137	025630	
7126	026322	005237	003324	
7127	026326	000207		
7128				
7129	026330			
(3)	026330			
(3)	026330	104401		

	INC	TRK(R5)	NO-SET FLAG FOR NEXT PASS
	MOV	SECREC,REC(R5)	GET NEW STARTING BLOCK #
	MOV	TAPLEN,TMP(R5)	RESET # OF BLOCKS
	JMP	1#	AND EXECUTE
TST8EX:	INC	DONE	DONE
	RETURN		RETURN
	ENDTST		

L10023: TRAP C#ETST


```

7185 026614 010110          MOV     R1,(R0)          ;INSERT INTO PACKET
7186                                ;SET UP EXPECTATIONS:
7187 026616 012765 000016 000070      MOV     #RSSNSZ,SND CNT(R5) ;HOW MANY TO SEND
7188 026624 112765 000002 000034      MOV     #RSCMND,XSFLG(R5) ;EXPECT END PACK
7189 026632 012765 000016 000036      MOV     #RSNDSZ,XSCNT(R5) ;COUNT WITH THIS
7190 026640 012765 000001 000032      MOV     #1.,XSPKNM(R5) ;EXPECT ONLY 1 PACKET
7191
7192 026646 004737 006600          CALL    RSVP            ;SEND
7193                                ;AND RETURN TO SCHEDULER
7194
7195 026652 032715 000010          BIT     #BIT3,#R5       ;RETRY (FLAG BYTE ERROR)?
7196 026656 001321                BNE     65#             ;YES
7197
7198 026660 012700 027746          66#:  MOV     #TRBUF,R0       ;-->(POINT TO) XMIT BUFFER
7199 026664 112710 000002          MOV     #RSCMND,#R0     ;FORM COMMAND MESSAGE PACK
7200 026670 112760 000012 000001      MOV     #RSMsiz,1(R0)   ;THIS BIG
7201 026676 112760 000000 000002      MOV     #RSSNOP,2(R0)   ;OP CODE IS NO-OPERATION
7202 026704 013760 000064 000012      MOV     REC,10.(R0)     ;TO THIS RECORD
7203 026712 105060 000003          CLR     3.(R0)         ;NO MODIFIER
7204 026716 105060 000004          CLR     4.(R0)         ;NO DRIVE
7205 026722 112760 000010 000005      MOV     #BIT3,5.(R0)   ;SET MRSP SWITCH
7206 026730 005060 000006          CLR     6.(R0)         ;NO SEQUENCE #
7207 026734 005060 000010          CLR     8.(R0)         ;NO BYTE COUNT
7208 025740 012701 000012          MOV     #RSMsiz,R1     ;GET COUNT
7209 026744 005721                TST     (R1)+           ;PLUS FLAG + BCNT
7210                                ;FOR CHECKSUM CALC
7211 026746 004737 013674          CALL    CHKSUM         ;R0-->TOP R1=# OF BYTES
7212 026752 010110          MOV     R1,(R0)       ;INSERT INTO PACKET
7213                                ;SET UP EXPECTATIONS:
7214 026754 012765 000016 000070      MOV     #RSSNSZ,SND CNT(R5) ;HOW MANY TO SEND
7215 026762 112765 000002 000034      MOV     #RSCMND,XSFLG(R5) ;EXPECT END PACK
7216 026770 012765 000016 000036      MOV     #RSNDSZ,XSCNT(R5) ;COUNT WITH THIS
7217 026776 012765 000001 000032      MOV     #1.,XSPKNM(R5) ;EXPECT ONLY 1 PACKET
7218
7219 027004 004737 006600          CALL    RSVP            ;SEND
7220                                ;AND RETURN TO SCHEDULER
7221
7222 027010 032715 000010          BIT     #BIT3,#R5       ;RETRY (FLAG BYTE ERROR)?
7223 027014 001321                BNE     66#             ;YES
7224
7225 027016 005237 003324          INC     DONE
7226 027022 005037 003344          CLR     TEST9
7227
7228 027026 005737 002224          TST     PPSOT9         ;PROTOCOL SUMMARY @ END OF PASS
7229 027032 001477                BEQ     ENOT9          ;NO
7230 027034 005037 027412          CLR     UNITNO        ;SET UNIT # TO ZERO
7231 027040          PRINTF #MSAGE1       ;PRINT HEADER
7232 027040 012746 027236          (7)   MOV     #MSAGE1.
7233 027044 012746 000001          (6)   MOV     #1,-(SP)
7234 027050 010600          (3)   MOV     SP,R0
7235 027052 104417          (4)   TRAP   C$PNTF
7236 027054 062706 000004          (4)   ADD     #4,SP
7237 027060 012737 003352 003314          MOV     #BLKTB,DEV PTR ;SET ALL UNITS
7238 027066 017705 154222          1#:  MOV     #DEV PTR,R5   ;GET POINTER
7239 027072 005765 000000          TST     STATUS(R5)    ;IS UNIT ABORTED
7240 027076 100431          BMI     3#             ;YES

```


E10

MISCELLANEOUS SECTIONS MACY11 30(1046) 25-JAN-84 08:33 PAGE 58-2
 CZTUUF.P11 25-JAN-84 08:09 TEST 9 / TESTS MODIFIED RADIAL SERIAL PROTOCOL

SEQ 0121

7236	027100	005765	000210		TST	MRSP(R5)		; IS UNIT MODIFIED		
7237	027104	001413			BEQ	2:		; NO		
7238	027106				PRINTF	#MSAGE2,UNITNO		; MESSAGE FOR MODIFIED UNIT		
(8)	027106	013746	027412						MOV	UNITNO,-
(7)	027112	012746	027277						MOV	#MSAGE2,
(6)	027116	012746	000002						MOV	#2,-(SP)
(3)	027122	010600							MOV	SP,RO
(4)	027124	104417							TRAP	C:PNTF
(4)	027126	062706	000006						ADD	#6,SP
7239	027132	000425			BR	4:		; SEE IF LAST UNIT		
7240	027134			2:	PRINTF	#MSAGE3,UNITNO		; MESSAGE FOR NON-MODIFIED UNIT		
(8)	027134	013746	027412						MOV	UNITNO,-
(7)	027140	012746	027333						MOV	#MSAGE3,
(6)	027144	012746	000002						MOV	#2,-(SP)
(3)	027150	010600							MOV	SP,RO
(4)	027152	104417							TRAP	C:PNTF
(4)	027154	062706	000006						ADD	#6,SP
7241	027160	000412			BR	4:		; SEE IF LAST UNIT		
7242	027162			3:	PRINTF	#MSAGE4,UNITNO		; MESSAGE FOR ABORTED UNIT		
(8)	027162	013746	027412						MOV	UNITNO,-
(7)	027166	012746	027362						MOV	#MSAGE4,
(6)	027172	012746	000002						MOV	#2,-(SP)
(3)	027176	010600							MOV	SP,RO
(4)	027200	104417							TRAP	C:PNTF
(4)	027202	062706	000006						ADD	#6,SP
7243	027206	023727	003314	003370	4:	CHP	DEVPTR,#LSTDEV	; IS THIS THE LAST DEVICE		
7244	027214	103006			BHIS	ENDT9		; YES		
7245	027216	062737	000002	003314	ADD	#2,DEVPTR		; GET NEXT UNIT		
7246	027224	005237	027412		INC	UNITNO		; INC UNIT #		
7247	027230	000716			BR	1:				
7248										
7249	027232	000207			ENDT9:	RETURN				
7250										
7251	027234				ENDTST					
(3)	027234									
(3)	027234	104401							L10024:	TRAP C:ETST
7252										
7253	027236	047045	051445	022470	MSAGE1:	.ASCIZ	/#N#S8#AUNIT NO#S9#S6#APROTOCOL#N/			
7254	027277	045	022516	034523	MSAGE2:	.ASCIZ	!#N#S9#S2#01#S9#S9#ARSP/MRSP!			
7255	027333	045	022516	034523	MSAGE3:	.ASCIZ	/#N#S9#S2#01#S9#S9#ARSP/			
7256	027362	047045	051445	022471	MSAGE4:	.ASCIZ	/#N#S9#S2#01#S9#S9#A---/			
7257		027412				.EVEN				
7258	027412	000000			UNITNO:	.WORD				

F10

MISCELLANEOUS SECTIONS MACY11 30(1046) 25-JAN-84 08:33 PAGE 59
CZTUUF.P11 25-JAN-84 08:09 PATCH AREA

SEQ 0122

7261
7262
7263
7264

000144

.SBTTL PATCH AREA
.REPT 100.
.WORD
.ENDR

```

7267
7268
7269
7270
7271 027724 031004
7272 027726 032042
7273 027730 033100
7274 027732 034136
7275 027734 035174
7276 027736 036232
7277 027740 037270
7278 027742 040326
7279
7280
7281
7282
7283
7284 027744 023
7285 027745 023
7286
7287 027746 001036
7288
7289
7290
7291 031004 001036
7292 032042 001036
7293 033100 001036
7294 034136 001036
7295 035174 001036
7296 036232 001036
7297 037270 001036
7298 040326 001036
7299
7300
7301
7302 041364

.SBTTL I/O BUFFER AREAS:
;WHO-GETS-WHAT-SPACE TABLE
BUFTBL: .WORD BUF0
        .WORD BUF1
        .WORD BUF2
        .WORD BUF3
        .WORD BUF4
        .WORD BUF5
        .WORD BUF6
        .WORD BUF7

;-----
;ONLY 1 TRANSMIT BUFFER NECESSARY:
        .BYTE RSXOFF
        .BYTE RSXOFF ;SEND XOFF BEFORE EVERY PACKET
TRBUF: .BLKB RCBFSZ
;-----

BUF0: .BLKB RCBFSZ
BUF1: .BLKB RCBFSZ
BUF2: .BLKB RCBFSZ
BUF3: .BLKB RCBFSZ
BUF4: .BLKB RCBFSZ
BUF5: .BLKB RCBFSZ
BUF6: .BLKB RCBFSZ
BUF7: .BLKB RCBFSZ

;-----
ENDMOD

```

```

7326          .TITLE PARAMETER CODING
7337
7338          .SBTTL  HARDWARE PARAMETER CODING SECTION
7366
7367 041364          BGNMOD
7368
7369
7370          ;**
7371          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
7372          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7373          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7374          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
7375          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7376          ; WITH THE OPERATOR.
7377          ;--
7378 041364          BGNHRD
7379 (3) 041364 000021          L$HARD: .WORD L10025-L$H
7380 (3) 041366
7381 041366          GPRMA  MSG1,0,0,160000,177777,YES
7382 (4) 041366 000031          .WORD  T$CODE
7383 (4) 041370 041430          .WORD  MSG1
7384 (4) 041372 160000          .WORD  T$LOLIM
7385 (4) 041374 177777          .WORD  T$HILIM
7386 041376          GPRMA  MSG1B,2,0,0,776,YES
7387 (4) 041376 001031          .WORD  T$CODE
7388 (4) 041400 041441          .WORD  MSG1B
7389 (4) 041402 000000          .WORD  T$LOLIM
7390 (4) 041404 000776          .WORD  T$HILIM
7391 041406          GPRML  MSG1C,6,1,YES
7392 (4) 041406 003130          .WORD  T$CODE
7393 (4) 041410 041456          .WORD  MSG1C
7394 (4) 041412 000001          .WORD  1
7395 041414          GPRML  MSG2,4,1,YES
7396 (4) 041414 002130          .WORD  T$CODE
7397 (4) 041416 041474          .WORD  MSG2
7398 (4) 041420 000001          .WORD  1
7399 041422          GPRML  MSG3,4,2,YES
7400 (4) 041422 002130          .WORD  T$CODE
7401 (4) 041424 041511          .WORD  MSG3
7402 (4) 041426 000002          .WORD  2
7403
7404          ENDHRD
7405 (2)
7406 (3) 041430          L10025: .EVEN
7407
7408 041430 052524 034065 041440  MSG1:  .ASCIZ  /TUSB CSR/
7409 041441 126 041505 047524  MSG1B: .ASCIZ  /VECTOR ADDR./
7410 041450 042120 020124 047111  MSG1C: .ASCIZ  /PDT INTERFACE/
7411 041474 042524 052123 042040  MSG2:  .ASCIZ  /TEST DRIVE 0/
7412 041511 124 051505 020124  MSG3:  .ASCIZ  /TEST DRIVE 1/
7413
7414          .EVEN

```

```

7411 .SBTTL SOFTWARE PARAMETER CODING SECTION
7412 ;**
7413 ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
7414 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
7415 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7416 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
7417 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7418 ; WITH THE OPERATOR.
7419 ;--
7420
7421 041526 BGNSFT
(3) 041526 000034
(3) 041530
7422 041530 GPRMD MSG4,0,D,1777,8.,512.,YES L$SOFT: .WORD L10026-L$S
(4) 041530 000052 .WORD T$CODE
(4) 041532 041620 .WORD MSG4
(4) 041534 001777 .WORD 1777
(4) 041536 000010 .WORD T$LOLIM
(4) 041540 001000 .WORD T$HILIM
7423 041542 GPRML MSG4B,10,1,YES
(4) 041542 004130 .WORD T$CODE
(4) 041544 041665 .WORD MSG4B
(4) 041546 000001 .WORD 1
7424 041550 GPRML MSG5,2,1,YES
(4) 041550 001130 .WORD T$CODE
(4) 041552 041727 .WORD MSG5
(4) 041554 000001 .WORD 1
7425 041556 GPRML MSG6,6,1,YES
(4) 041556 003130 .WORD T$CODE
(4) 041560 041761 .WORD MSG6
(4) 041562 000001 .WORD 1
7426 041564 GPRML MSG7,4,1,YES
(4) 041564 002130 .WORD T$CODE
(4) 041566 042006 .WORD MSG7
(4) 041570 000001 .WORD 1
7427 041572 GPRMD MSG8,10.,D,377,1,254.,YES
(4) 041572 005052 .WORD T$CODE
(4) 041574 042034 .WORD MSG8
(4) 041576 000377 .WORD 377
(4) 041600 000001 .WORD T$LOLIM
(4) 041602 000376 .WORD T$HILIM
7428 041604 GPRML MSG9,12.,1,YES
(4) 041604 006130 .WORD T$CODE
(4) 041606 042075 .WORD MSG9
(4) 041510 000001 .WORD 1
7429 041612 GPRML MSG10,14.,1,YES
(4) 041612 007130 .WORD T$CODE
(4) 041614 042142 .WORD MSG10
(4) 041616 000001 .WORD 1
7436 041620 SFTOUT: ENDSFT
(2)
(3) 041620 L10026: .EVEN
7437 041620 052516 041115 051105 MSG4: .ASCIZ 'NUMBER OF BLOCKS:TEST 5-8 (8 TO 512)
7438 041665 101 042104 042040 MSG4B: .ASCIZ /ADD DR # TO DATA PAT'ERN:TEST 5-8/
7439 041707 123 040524 044524 MSG5: .ASCIZ /STATISTICS PRINTED AT EOP/
7440 041761 103 046517 040520 MSG6: .ASCIZ /COMPARE DATA ON READ/

```

PARAMETER CODING MACY11 30(1046) 25-JAN-84 08:33 PAGE 62-1
CZTUUF.P11 25-JAN-84 08:09 SOFTWARE PARAMETER CODING SECTION

SEQ 0126

7441	042006	051120	047111	020124
7442	042034	020043	051105	047522
7443	042075	120	044522	052116
7444	042142	042524	052123	047440
7445				

MSG7:	.ASCIZ	/PRINT PACKET ON ERROR/
MSG8:	.ASCIZ	/ * ERRORS = DVC FATAL IF 'EVL' SET /
MSG9:	.ASCIZ	/PRINT UNIT PROTOCOL SUMMARY (TEST 9) /
MSG10:	.ASCIZ	/TEST ONLY DRIVE 0 IN TEST 3 /
	.EVEN	

7448		000016		.REPT 14.		;LASTAD CORRECTION
7449				.WORD		
7450				.ENDR		
7457	042232			LASTAD		
(2)						.EVEN
(2)	042232	042252				.WORD T\$FREE
(2)	042234	000006				.WORD T\$SIZE
(3)	042236		L\$LAST::			
7458	042236			ENDMOD		
7459						
7460	042236			BGNSETUP	1	
7461	042236			BGNPTAB		
(4)	042236	000000				.WORD 0
(3)	042240	000004				.WORD L10031-
(3)	042242					L10027:
7462	042242	176500		176500		
7463	042244	000300		300		
7464	042246	000003		3		
7465	042250	000000		0		
7466	042252			ENDPTAB		
(3)	042252					L10031:
7467	042252			ENDSETUP		
7468		000001	.END			

ENDSW	978	3772	3900														
ENOTST	992	3772	6900	6927	6962	7004	7036	7067	7098	7129	7251						
EQUALS	1015	3772	3962														
ERRDF	1093	3772	6011	6025													
ERRHRD	1105	3772	6038	6053													
ERROR	1115	3772															
ERRSF	1124	3772	5147	6574	6594												
ERRSOF	1136	3772	6031	6049	6362												
ERRTBL	1146	3772															
ESCAPE	1161	3772															
EXIT	1191	3772	6893	6907	6937	6972	7012	7043	7074	7105	7141						
FEQUAL	1233	3772															
GETBYT	1251	3772															
GETPRI	1269	3772															
GETWOR	1261	3772															
GMANIA	1291	3772															
GMANID	1304	3772															
GMANIL	1320	3772															
GPHARD	1333	3772	6581														
GPRMA	1345	3772	7381	7382													
GPRMD	1377	3772	7422	7427													
GPRML	1412	3772	7383	7384	7385	7423	7424	7425	7426	7428	7429						
HEADER	1437	3772	3815														
INLOOP	1451	3772															
IOSETU	1458	3772															
IOSTAR	1471	3772															
KT11	1493	3772															
LASTAD	1664	3772	7457														
MANUAL	1682	3772															
MEMORY	1690	3772															
M\$BYTE	2912	3772	3815														
M\$CHEC	3218	3772	6893	6907	6937	6972	7012	7043	7074	7105	7141						
M\$CNTD	3291	3772	7381	7382	7383	7384	7385	7422	7423	7424	7425	7426	7427	7428	7429		
M\$COUN	3136	3772	5502	5823	5828	5831	5833	5847	5854	5862	6077	6080	6083	6363	6397		
	6402	3772	6477	6479	6495	6496	6506	6701	6761	7231	7238	7240	7242				
M\$DATA	2625	3772	3815	3817	4334												
M\$DECR	3075	3772	3828	3873	3900	3902	6087	6291	6304	6416	6519	6535	6662	6693	6743		
	6781	3772	6900	6927	6962	7004	7036	7067	7098	7129	7251	7302	7393	7436	7458		
	7461																
M\$DEFA	3275	3772	7381	7382	7383	7384	7385	7422	7423	7424	7425	7426	7427	7428	7429		
M\$ENDE	3157	3772	3873	3900	3902	6087	6291	6304	6416	6519	6535	6662	6693	6743	6781		
	6826	3772	6900	6927	6962	7004	7036	7067	7098	7129	7251	7302	7393	7436	7458		
M\$ERRI	2372	3772	5147	6011	6025	6031	6038	6049	6053	6362	6574	6594					
M\$ESCA	2932	3772															
M\$ESCS	2945	3772															
M\$EXCP	3198	3772	7381	7382	7422	7427											
M\$EXIT	2954	3772	6893	6907	6937	6972	7012	7043	7074	7105	7141						
M\$EXSE	2976	3772	6893	6907	6937	6972	7012	7043	7074	7105	7141						
M\$EXTJ	2965	3772	6893	6907	6937	6972	7012	7043	7074	7105	7141						
M\$GEN	3099	3772	3815	3817	3824	3843	3860	3873	3882	3900	4334	6070	6087	6287	6291		
	6295	3772	6304	6467	6519	6545	6662	6679	6693	6716	6743	6753	6781	6803	6826	6891	
	6900	3772	6905	6927	6935	6962	6970	7004	7010	7036	7041	7067	7072	7098	7103	7129	
	7134	3772	7251	7378	7393	7421	7436	7457	7461	7466							
M\$GENB	2775	3772															
M\$GETS	3091	3772	3828	3873	3900	3902	6087	6291	6304	6416	6519	6535	6662	6693	6743		
	6781	3772	6900	6927	6962	7004	7036	7067	7098	7129	7251	7302	7393	7436	7458		

MIGETT	2645#	3772#	6893#	6907#	6937#	6972#	7012#	7043#	7074#	7105#	7141#				
MIGNGB	2700#	3772#	3798#	3815#	3817#	3824#	3843#	3860#	3882#	3955#	4334#	6070#	6287#	6295#	6460#
	6467#	6545#	6679#	6716#	6753#	6803#	6888#	7367#	7478#	7421#	7457#				
MIGNIN	3113#	3772#	3815#	3817#	3843#	3860#	3882#	4334#	5060#	5147#	5148#	5150#	5271#	5502#	5595#
	5823#	5828#	5831#	5833#	5847#	5854#	5862#	6011#	6025#	6031#	6038#	6049#	6053#	6057#	6077#
	6080#	6083#	6087#	6222#	6237#	6238#	6240#	6278#	6280#	6291#	6304#	6312#	6313#	6362#	6363#
	6397#	6402#	6404#	6475#	6477#	6478#	6479#	6480#	6495#	6496#	6506#	6519#	6553#	6554#	6556#
	6574#	6575#	6581#	6582#	6594#	6595#	6632#	6662#	6681#	6692#	6693#	6701#	6704#	6721#	6742#
	6743#	6761#	6781#	6826#	6893#	6900#	6907#	6927#	6937#	6962#	6972#	7004#	7012#	7036#	7043#
	7067#	7074#	7098#	7105#	7129#	7141#	7231#	7238#	7240#	7242#	7251#	7378#	7381#	7382#	7383#
	7384#	7385#	7393#	7421#	7422#	7423#	7424#	7425#	7426#	7427#	7428#	7429#	7436#	7457#	7461#
MIGNLS	2728#	3772#													
MIGNSU	2690#	3772#													
MIGNTA	2670#	3772#	3873#	3900#	6087#	6291#	6304#	6519#	6662#	6693#	6743#	6781#	6826#	6900#	6927#
	6962#	7004#	7036#	7067#	7098#	7129#	7231#	7393#	7436#	7461#	7466#				
MIGNTE	2680#	3772#	6891#	6905#	6935#	6970#	7010#	7041#	7072#	7103#	7134#				
MIHAPT	2484#	3772#	3815#												
MHNAP	2577#	3772#	3815#												
MINCR	3066#	3772#	3798#	3824#	3860#	3882#	3955#	5060#	5147#	5148#	5150#	5271#	5502#	5595#	5823#
	5828#	5831#	5833#	5847#	5854#	5862#	6011#	6025#	6031#	6038#	6049#	6053#	6057#	6070#	6077#
	6080#	6083#	6087#	6222#	6237#	6238#	6240#	6278#	6280#	6287#	6295#	6312#	6313#	6362#	6363#
	6397#	6402#	6404#	6460#	6467#	6475#	6477#	6478#	6479#	6480#	6495#	6496#	6506#	6519#	6545#
	6553#	6556#	6574#	6575#	6581#	6594#	6595#	6632#	6662#	6679#	6681#	6692#	6693#	6701#	6704#
	6716#	6721#	6742#	6743#	6753#	6761#	6791#	6803#	6826#	6888#	6891#	6893#	6900#	6905#	6907#
	6927#	6935#	6937#	6962#	6970#	6972#	7004#	7010#	7012#	7036#	7041#	7043#	7067#	7072#	7074#
	7098#	7103#	7105#	7129#	7134#	7141#	7231#	7238#	7240#	7242#	7251#	7367#	7378#	7421#	7460#
	7461#														
MIOSE	2438#	3772#													
MILDRO	2782#	3772#	6057#	6237#	6278#	6280#	6553#	6581#	6692#	6704#					
MIMASK	2397#	3772#													
MIMCHI	90#	3772#													
MIMCLO	213#	3772#													
MIMSK1	240#	3772#													
MIPOP	265#	3772#	3828#	3873#	3900#	3902#	6087#	6291#	6304#	6416#	6519#	6535#	6662#	6693#	6743#
	6781#	6826#	6900#	6927#	6962#	7004#	7036#	7067#	7098#	7129#	7251#	7302#	7393#	7436#	7458#
MIPRIN	2356#	3772#	5502#	5823#	5828#	5831#	5833#	5847#	5854#	5862#	6077#	6080#	6083#	6363#	6397#
	6402#	6404#	6477#	6479#	6495#	6496#	6506#	6701#	6761#	7231#	7238#	7240#	7242#		
MIPUSH	2344#	3772#	3798#	3824#	3860#	3882#	3955#	6070#	6287#	6295#	6460#	6467#	6545#	6679#	6716#
	6753#	6803#	6888#	6891#	6905#	6935#	6970#	7010#	7041#	7072#	7103#	7134#	7367#	7378#	7421#
MIPUT	2830#	3772#	5502#	5823#	5828#	5831#	5833#	5847#	5854#	5862#	6077#	6080#	6083#	6238#	6240#
	6363#	6397#	6402#	6404#	6477#	6479#	6495#	6496#	6506#	6681#	6701#	6761#	7231#	7238#	7240#
	7242#														
MIPUT1	2853#	3772#	5502#	5823#	5828#	5831#	5833#	5847#	5854#	5862#	6077#	6080#	6083#	6238#	6240#
	6363#	6397#	6402#	6404#	6477#	6479#	6495#	6496#	6506#	6681#	6701#	6761#	7231#	7238#	7240#
	7242#														
MIRADI	3163#	3772#	7381#	7382#	7383#	7384#	7385#	7422#	7423#	7424#	7425#	7426#	7427#	7428#	7429#
MIRBRO	2796#	3772#													
MIRNRO	2813#	3772#	6581#	6632#											
MISETS	3083#	3772#	3798#	3824#	3860#	3882#	3955#	6070#	6287#	6295#	6460#	6467#	6545#	6679#	6716#
	6753#	6803#	6888#	6891#	6905#	6935#	6970#	7010#	7041#	7072#	7103#	7134#	7367#	7378#	7421#
MISTAR	2475#	3772#													
MISVC	2757#	3772#	5060#	5147	5148#	5150#	5271#	5502#	5595#	5823#	5828#	5831#	5833#	5847#	5854#
	5862#	6011	6025	6031	6038	6049	6053	6057#	6077#	6080#	6083#	6087#	6222#	6237#	6238#
	6240#	6278#	6280#	6312#	6313#	6362	6363#	6397#	6402#	6404#	6475#	6477#	6478#	6479#	6480#
	6495#	6496#	6506#	6519#	6553#	6556#	657	6575#	6581#	6594	6595#	6632#	6662#	6681#	6692#
	6693#	6701#	6704#	6721#	6742#	6743#	6761#	6781#	6826#	6893#	6900#	6907#	6927#	6937#	6962#

D12

PARAMETER CODING MACY11 30(1046) 25-JAN-84 08:33 PAGE 65-4
CZTUUF.P11 25-JAN-84 08:09 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0146

ERRORS DETECTED: 0

CZTUUF.BIN/EN:AMA:ABS.CZTUUF/CRF=SVC.SML.CZTUUF.P11
RUN-TIME: 19 24 2 SECONDS
RUN-TIME RATIO: 68/46=1.4
CORE USED: 23K (46 PAGES)

