.REM &

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

HISTORY

JUNE 18,1979	INITIAL RELEASE	CZTUUA0
JULY 1,1979	SECOND RELEASE	CZTUUB0
JUNE 1,1980	THIRD RELEASE	CZTUUB1
OCTOBER 1,1981	FOURTH RELEASE	CZTUUC0
MARCH 1,1982	FIFTH RELEASE	CZTUUD0
JUNE 1,1983	SIXTH RELEASE	CZTUUE0
JANUARY 23,1984	SEVENTH RELEASE	CZTUUF0

CZTUUA0

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

CHANGES TO CZTUUA0

1. THE PROGRAM WAS MODIFIED TO RUN UNDER THE NEW DIAGNOSTIC SUPERVISOR CHSAA0. AS A RESULT OF THIS CONVERSION, THIS PROGRAM NOW OPERATES IN 8K AND PAPERTAPE DISTRIBUTION REQUIRES ONLY ONE PART AK-E6508-MC.

CHANGES TO CZTUUB0

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 CZTUUB1

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

CHANGES TO CZTUUC0

1. A TEST WAS ADDED TO VERIFY 128 BYTE/BLOCK MODE. THE TEST IS SIMULAR TO TEST 3. IT WRITES, READS, AND VERIFIES SEQUENTIAL BLOCKS OF TAPE FROM BLOCK 0 THOUGH 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.

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.

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

THIS DIAGNOSTIC EXERCISES FROM 1 TO 8 TU58 CONTROLLER BOARDS, EACH OF WHICH MAY SUPPORT 1 OR 2 DRIVES. THE PROGRAM IMPLEMENTS THE "MAINTENANCE MODE" SWITCH WITHIN ALL PACKET COMMANDS, THUS RETRIEVING MAXIMUM INFORMATION FROM THE DEVICE UPON CERTAIN DEVICE RECOGNIZED ERRORS.

STATISTICAL SUMMARIES ARE PROVIDED FOR ALL UNITS TESTED. RETRIES ARE PERFORMED ON DATA-RELATED ERROR CONDITIONS.

USE OF LOOP ON ERROR FLAG (:LOE) IS IMPLEMENTED BUT NOT RECOMMENDED FOR USE, SINCE THE LOOPS ARE QUITE LENGTHLY DUE TO COMMUNICATIONS PROTOCOL OVERHEAD.

1.1 PROGRAM ABSTRACT

IN ORDER TO EXERCISE MULTIPLE UNITS IN AN EFFICIENT MANNER, A SCHEDUL.NG ALGORITHM BUILDS, THEN SENDS THE NEXT COMMUNICATION PACKET (COMMAND OR DATA) FORMULATED BY EXECUTING MACRO CODE WITHIN THE TEST ALGORITHMS. THE USE OF MACROS TO IMPLEMENT THE COMMUNICATIONS PROTOCOL SIMPLIFIES CONTEXT SWITCHING FROM UNIT TO UNIT BY NOT REQUIRING 8 SEPARATE DEVICE STACKS IN ADDITION TO THE SYSTEM STACK. THE TEST CODE RUNS AS A CO-ROUTINE WITH THE SCHEDULER,SO A TEST CODE PROGRAM COUNTER IS MAINTAINED FOR EACH UNIT "TSTPC(R5)".

THE TESTS ARE PERFORMED USING THE SPECIFIED ALGORITHM ON ALL DRIVE 0'S, THEN REPEAT THE TEST AFTER SWITCHING DRIVES, IF ANY DRIVE "1'S" WERE SELECTED.

FOLLOWING THE TRANSMISSION OF 1 PACKET TO EACH DEVICE (WITH XOFF PRECEEDING) THE UNITS ARE POLLED, AND THEIR ENTIRE RESPONSES

EVALUATED ROUND ROBIN. IF ANY ERROR INITIATES A RETRY, THE SCHED ULING PROCESS IS MODIFIED TO COMMUNICATE WITH ONLY 1 UNIT UNTIL COMPLETION OF THE RETRY PROCEDURE. THEN, A RETRY BY ANOTHER UNIT MAY PROCEED, OR THE SYSTEM CONTINUES NORMALLY.

THROUGHOUT THE PROGRAM, R5 POINTS TO ONE OF 8 POSSIBLE DATA STRUCTURES CONTAINING STATUS, TEST PARAMETERS, AND STATISTICAL INFORMATION FOR THE CURRENT UNIT, CALLED "UNIT'S DATA BLOCK". "START" CLEARS STATISTICS. "RESTART" AND "CONTINUE" DO NOT.

UPON OCCURANCE OF A FATAL ERROR, THAT UNIT IS DESCHEDULED (ABORTED) ALLOWING THE REMAINING (IF ANY) TO PROCEED WITH TESTING.

ERROR DESCRIPTIONS:

AN EXPLANATION OF THE EXTENDED ERROR INFORMATION FOLLOWS. SEE ALSO THE SECTION IN THIS LISTING SUBTITLED "ERROR MESSAGE DESCRIPTIONS".

BLOCK #:	THE RECORD NUMBER (1 PER 512. BYTES) IN LAST COMMAND PACK.
COMMAND:	THE MOST RECENT COMMAND PACKET OP CODE.
EXPCTD:	THE DATA PATTERN USED ON WRITE COMMAND AND FOR DATA COMPARE AFTER READ OP.
SUCCESS:	THE SUCCESS CODE RECEIVED IN END PACKET.
PAK SENT:	TYPE OF PACKET JUST SENT (0 FOR DATA; 1 FOR COMMAND)
FLAG RCVD:	FLAG BYTE OF PACKET CURRENTLY BEING CHECKED, OR 1ST BYTE OF RESPONSE.

SINCE IN MAINTENANCE MODE TU58 WILL SEND A BAD DATA PACK WITH A "DATA CHECK" SUCCESS STATUS IN THE FOLLOWING END PACK, THE HOST WILL, UPON CHECKING THOSE DATA PACK(S), DETERMINE "BAD DATA" IN PACKET ERROR FIRST, THEN INTERPRET THE SUCCESS CODE TO DIFFERENTIATE A COMMUNICATIONS GLITCH (GOOD SUCCESS) VS. TU 'DATA-CHECK' ERROR CODE. THIS WOULD SEEM TO RESULT IN TWO "ERROR" MESSAGES FOR ONE ERROR CONDITION, BUT ONLY THE SECOND ERROR MESSAGE WILL CONTAIN PERTINENT (NOT ZERO) ERROR NUMBER.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE

PDP-11/LSI-11 CPU WITH AT LEAST 24K WORDS OF MEMORY AND CONSOLE DEVICE.

TU58 CONTROLLER AND DRIVE(S). DL, DLV, OR PDT COMPATABLE INTER FACE; AND REVISION "I" TU58 MICROCODE (OR LATER) ASSUMED.

1.2.2 SOFTWARE

THE PROGRAM IS REVISION D DIAGNOSTIC SUPERVISOR COMPATIBLE. CONSULT XXDP+ USERS MANUAL FOR OPERATING INSTRUCTIONS.

1.3 RELATED DOCUMENTS AND STANDARDS

XXDP+ USERS MANUAL CHQUS

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

APPROPRIATE INTERFACE DIAGNOSTICS MAY BE RUN TO ISOLATE INTERFACE ERRORS.

1.5 ASSUMPTIONS

SYSTEM HARDWARE OTHER THAN TU58(S) IS OPERATIONAL.

2.0 OPERATING INSTRUCTIONS

2.1 HOW TO RUN THIS DIAGNOSTIC

THE DIAGNOSTIC MAY BE INVOLKED WITH A 'START' RESPONSE TO THE SUPERVISOR PROMPT. 'STA'(CR) IS SUFFICIENT.
IF THE DEVICE IS NOT AT THE STANDARD ADDRESS AND VECTOR (176500. 300), THEN ANSWER "CHANGE HW?" WITH 'YES' INITIALLY TO SET UP HARDWARE CONFIGURATION TABLES FOR EACH UNIT. THAT INFORMATION IS:

TU58 CSR - ADDRESS OF RCSR OF DLV-11 OR OTHER INTERFACE BOARD.

VECTOR ADDR. - ADDRESS OF INTERRUPT VECTOR LOCATION.

PDT INTERFACE -- IS THE TU58 IN A PDT 11/130,
OR SYSTEM WHOSE BUFFERS ARE:
RCSR
RCDB (AND XMDB)
XMSR

TEST DRO YES OR NO

TEST DR1 - YES OR NO

SUBSEQUENT RESPONSES TO "CHANGE HW?" MAY THEN BE 'NO".

THE STANDARD ADDRESS AND VECTOR LOCATIONS FOR THE PDT 11/130

ARE 177170 AND 260 RESPECTIVELY.

THE SOFTWARE QUESTIONS ARE AS FOLLOWS:

NUMBER OF BLOCKS: TEST 5-8 -- ONE MAY SELECT A MINIMUM OF 8, TO A MAXIMUM OF 512 BLOCKS TO WRITE, READ; WRITE VERIFY; AND READ REDUCED, AS EXPLAINED IN SECTION 6.0.

ADD DR # TO DATA PATTERN -- FOR THOSE SAME READ AND WRITE TESTS 5-8, THE DRIVE NUMBER (0 OR 1) MAY BE ADDED TO DATA WRITTEN ON TAPE TO INSURE DRIVE SELECT BIT OPERATION.

STATISTICS PRINTED AT EOP -- SELECTS WHETHER OR NOT TO PRINT INFORMATION AT END OF PASS OR ↑C. THESE STATISTICS MAY ALSO BE RE TRIEVED WITH THE "PRI" COMMAND.

COMPARE DATA ON READ -- SELECTS WHETHER OR NOT TO DO A DATA COMPARE ON DATA PACKETS RE CEIVED.

PRINT PACKET ON ERROR -- PRINTS 132. BYTE DATA PACKET ON A COMPARE ERROR,IF SELECTED.

ERRORS=DVC FATAL IF 'EVL' SET -- IF USER SETS EVL FLAG (EVALUATE) MODE), HRD OR SFT ERROR MESSAGES BECOME DVC FTL ERRORS AFTER THE NUMBER SPECIFIED IS EXCEEDED.

PRINT UNIT PROTOCOL SUMMARY (TEST 9) -- PRINTS A TABLE INDICATING THE PROTOCOL OF EACH UNIT.

3.0 ERROR INFORMATION

ERROR INFORMATION IS PROVIDED ON OCCURRANCE OF ERRORS AS OUTLINED IN SECTION 1.1.

4.0 PERFORMANCE AND PROGRESS REPORTS

STATISTICS ARE AVAILABLE PER SECTION 1.1 AT END OF PASS, CONTROL C, OR UPON ENTERING A "PRI" COMMAND. THEY CONSIST OF # BLOCKS WRITTEN AND READ, # OF DATA ERRORS, HARD OR SOFT.

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:

1. INIT CODE IN SUPERVISOR IS EXECUTED

OR

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 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 8. (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 TU58 A GET CHARACTERISTICS COMMAND AND MONITORING THE RESPONSE. IF THE TU58 RETURNS AN END PACKET IT IS A MODIFIED UNIT. IF THE TU58 RETURNS A DATA PACKET IT IS A NON-MODIFIED UNIT. NOTE, THE DATA PACKET RETURNED ON A GET CHARACTERISTICS COMMAND IS NOT NORMAL, RATHER IT CONSISTS OF A DATA PACKET THAT IS 28. BY"ES PLUS AN END PACKET WHICH IS 14. BYTES. THE SECOND PART OF TEST 9 TESTS ONLY THOUGHS 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 ADVAILABLE BY ANSWERING YES (Y) TO SOFTWARE (SW) QUESTION # 5.

```
                                .TITLE PROGRAM HEADER AND TABLES
                                .SBTTL  PROGRAM HEADER

                                        .ENABL  ABS,AMA
002000                                  .       =       2000
                                        .NLIST  BEX
002000                                  BGNMOD

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

002000                                  POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU,BGNSETUP


002000                                  HEADER  CZTUU,F,0,3600.,1,PR107
(4) 002000                                                              L$NAME::
(4) 002000  103                                                                 .ASCII /C/
(4) 002001  132                                                                 .ASCII /Z/
(4) 002002  124                                                                 .ASCII /T/
(4) 002003  125                                                                 .ASCII /U/
(4) 002004  125                                                                 .ASCII /U/
(6) 002005  000                                                                 .BYTE   0
(6) 002006  000                                                                 .BYTE   0
(5) 002007  000                                                                 .BYTE   0
(5) 002010                                                              L$REV::
(4) 002010  106                                                                 .ASCII  /F/
(5) 002011                                                              L$DEPO::
(4) 002011  060                                                                 .ASCII  /0/
(5) 002012                                                              L$UNIT::
(4) 002012  000001                                                              .WORD   T$PTHV
(5) 002014                                                              L$TIML::
(4) 002014  007020                                                              .WORD   3600.
(5) 002016                                                              L$HPCP::
(4) 002016  041366                                                              .WORD   L$HARD
(5) 002020                                                              L$SPCP::
(4) 002020  041530                                                              .WORD   L$SOFT
(5) 002022                                                              L$HPTP::
(4) 002022  002176                                                              .WORD   L$HW
(5) 002024                                                              L$SPTP::
(4) 002024  002210                                                              .WORD   L$SW
(5) 002026                                                              L$LADP::
(4) 002026  042236                                                              .WORD   L$LAST
(5) 002030                                                              L$STA::
(4) 002030  000000                                                              .WORD   0
(5) 002032                                                              L$CO::
(4) 002032  000000                                                              .WORD   0
(5) 002034                                                              L$DTYP::
(4) 002034  000001                                                              .WORD   1
(5) 002036                                                              L$APT::
(4) 002036  000000                                                              .WORD   0
(5) 002040                                                              L$DTP::
(4) 002040  002152                                                              .WORD   L$DISPAT
(5) 002042                                                              L$PRIO::
(4) 002042  000340                                                              .WORD   PRIO7
```

```
   (5)  002044
   (4)  002044  000000                                                  L$ENVI::
   (5)  002046                                                                  .WORD   0
                                                                        L$EXP1::
   (4)  002046  000000                                                          .WORD   0
   (5)  002050                                                          L$MREV::
   (4)  002050     003                                                          .BYTE   C$REVISI
   (3)  002051     003                                                          .BYTE   C$EDIT
   (5)  002052                                                          L$EF::
   (4)  002052  000000                                                          .WORD   0
   (5)  002054  000000                                                          .WORD   0
   (5)  002056                                                          L$SPC::
   (4)  002056  000000                                                          .WORD   0
   (5)  002060                                                          L$DEVP::
   (4)  002060  005512                                                          .WORD   L$DVTYP
   (5)  002062                                                          L$REPP::
   (4)  002062  015170                                                          .WORD   L$RPT
   (5)  002064                                                          L$EXP4::
   (4)  002064  000000                                                          .WORD   0
   (5)  002066                                                          L$EXP5::
   (4)  002066  000000                                                          .WORD   0
   (5)  002070                                                          L$AUT::
   (4)  002070  017326                                                          .WORD   L$AU
   (5)  002072                                                          L$DUT::
   (4)  002072  017202                                                          .WORD   L$DU
   (5)  002074                                                          L$LUN::
   (4)  002074  000000                                                          .WORD   0
   (5)  002076                                                          L$DESP::
   (4)  002076  002122                                                          .WORD   L$DESC
   (5)  002100                                                          L$LOAD::
   (4)  002100  104035                                                          EMT     E$LOAD
   (5)  002102                                                          L$ETP::
   (4)  002102  000000                                                          .WORD   0
   (5)  002104                                                          L$ICP::
   (4)  002104  016204                                                          .WORD   L$INIT
   (5)  002106                                                          L$CCP::
   (4)  002106  017160                                                          .WORD   L$CLEAN
   (5)  002110                                                          L$ACP::
   (4)  002110  016776                                                          .WORD   L$AUTO
   (5)  002112                                                          L$PRT::
   (4)  002112  002142                                                          .WORD   L$PROT
   (5)  002114                                                          L$TEST::
   (4)  002114  000000                                                          .WORD   0
   (5)  002116                                                          L$DLY::
   (4)  002116  000000                                                          .WORD   0
   (5)  002120                                                          L$HIME::
   (4)  002120  000000                                                          .WORD   0
 3816
 3817  002122                          DESCRIP <TU58 PERF EXER>
  (4)  002122                                                         L$DESC::
  (3)  002122  052524  034065  050040                                          .ASCIZ  /TU58 PE
  (3)  002130  051105  020106  054105
  (3)  002136  051105     000
  (2)          002142                                                          .EVEN
```

```
                                        ;**
                                        ;THE PROTECT TABLE IS USED BY THE MONITOR TO WARN THE OPERATOR WHEN HE
                                        ;TRIES TO TEST THE LOAD DEVICE.
                                        ;-

002142                                  BGNPROT
002142                                                                  L$PROT::
002142  000000                                  .WORD   0               ;DEVICE CSR
002144  177777                                  .WORD   -1              ;NO MASS BUS
002146  177777                                  .WORD   -1              ;NO DRIVE
002150                                  ENDPROT
```

```
3835
3836                                    .SBTTL  DISPATCH TABLE
3837
3838                                    ;++
3839                                    ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
3840                                    ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
3841                                    ;--
3842
3843  002150                                    DISPATCH 9
 (4)  002150  000011                                                        .WORD   9
 (3)  002152                                                        L$DISPATCH::
 (6)  002152  017330                                                        .WORD   T1
 (6)  002154  017532                                                        .WORD   T2
 (6)  002156  020004                                                        .WORD   T3
 (6)  002160  021376                                                        .WORD   T4
 (6)  002162  023002                                                        .WORD   T5
 (6)  002164  023772                                                        .WORD   T6
 (6)  002166  024556                                                        .WORD   T7
 (6)  002170  025546                                                        .WORD   T8
 (6)  002172  026332                                                        .WORD   T9
3844
```

```
                                        .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.
                                        ;-

002174                                          BGNHW   DFPTBL
(3)  002174  000004                                                             .WORD   L10001-L
(3)  002176                                                             L$HW::
(3)  002176                                                             DFPTBL::

002176  176500                                  .WORD   176500          ;CSR ADDRESS
002200  000300                                  .WORD   300             ;VECTOR ADDR.
002202  000003                                  .WORD   3               ;TEST DRIVE ZERO AND ONE
002204  000000                                  .WORD   0               ;NOT PDT TYPE INTERFACE

002206                                          ENDHW
(3)  002206                                                             L10001:
```

```
                                        .SBTTL  SOFTWARE P-TABLE

                                        ;++
                                        ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
                                        ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                                        ;--

002206                                          BGNSW   SFPTBL
(3)  002206  000010                                                             .WORD   L10002-L
(3)  002210                                                             L$SW::
(3)  002210                                                             SFPTBL::

002210  000010                          LENGTH: .WORD   8.              ;TAPE LENGTH
002212  000001                          STAEOP: .WORD   1               ;PRINT STATISTICS AT EOP
002214  000001                          PRBUF:  .WORD   1               ;PRINT DATA BUF ON COMP. ERROR
002216  000001                          CMPDAT: .WORD   1               ;COMPARE DATA
002220  000001                          DRVCHK: .WORD   1               ;ADD DR # TO DATA
002222  000001                          EVLTHR: .WORD   1               ;THRESHOLD FOR EVL TEST
002224  000000                          PPSOT9: .WORD   0               ;PRINT UNIT PROTOCOL SUMMARY (TST9)
002226  000000                          DOT3FL: .WORD   0               ;TEST 3-DRIVE 0 ONLY FLAG

002230                                          ENDSW
(3)  002230                                                             L10002:

002230                                          ENDMOD
```

```
                                        .TITLE GLOBAL AREAS
                                        .SBTTL  GLOBAL EQUATES SECTION


002230                                          BGNMOD

                                        ;++
                                        ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
                                        ; ARE USED IN MORE THAN ONE TEST.
                                        ;--

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
```

```
   (1)          000300                      PRIO6== 300
   (1)          000240                      PRIO5== 240
   (1)          000200                      PRIO4== 200
   (1)          000140                      PRIO3== 140
   (1)          000100                      PRIO2== 100
   (1)          000040                      PRIO1== 40
   (1)          000000                      PRIO0== 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
```

```
                                        .SBTTL  ERROR CODE EQUATES

                                        ;THE ERROR CODE OFFSET VALUES :
                                        ;USED BY ROUTINE 'LOG' TO INDEX (BY R5) INTO DEVICE'S DATA BLOCK AND
                                        ;INCREMENT STATISTICS.

000002                                  SFTRD   ==      2
000004                                  SFTWR   ==      4
000006                                  RCINIT  ==      6
000010                                  OTL     ==      8.
000012                                  OVRN    ==      10.
000014                                  BDCOM   ==      12.
000016                                  HRDRD   ==      14.
000020                                  HRDWR   ==      16.
000022                                  BDCHK   ==      18.
000024                                  SKERR   ==      20.
000026                                  WRLOCK  ==      22.
000030                                  NOMOT   ==      24.
000032                                  CNINIT  ==      26.
000034                                  PARTL   ==      28.
000036                                  NOUNIT  ==      30.
000040                                  CMNDER  ==      32.
000042                                  RECERR  ==      34.
000044                                  SLFER   ==      36.
000046                                  SUCOTL  ==      38.
000050                                  TORCVB  ==      40.
000054                                  NCART   ==      44.
000056                                  TOSNDB  ==      46.

                                        ;               IN ADDITION, SYSTEM SETUP OR RUNTIME ERRORS ARE:

                                        ;       100.    -       ALL UNITS ABORTED

                                        ;       101.    -       MORE THAN 8. UNITS (16 DRIVES) REQUESTED

                                        ;       102.    -       NEITHER DRIVE SELECTED FOR THIS CONTROLLER

                                        ;               ALL THE ABOVE ARE CLASSIFIED AS SYSTEM FATAL
```

```
                                .SBTTL  GENERAL EQUATES
                                ;RADIAL SERIAL CODES:
                                ;-------------------------------------- ------ ------  ----- -- -----
                                ;THE FLAG BYTE CODES ARE:
000002                                  RSCMND == 2             ;"COMMAND" PACKET
000020                                  RSCONT == 20            ;"CONTINUE" SINGLE BYTE
000020                                  RSXON  == 20            ;"XON" SINGLE BYTE
000023                                  RSXOFF == 23            ;"XOFF" SINGLE BYTE
000004                                  RSINIT == 4             ;"INIT" SINGLE BYTE
000001                                  RSDATA == 1             ;"DATA" PACKET
000002                                  RSEND  == RSCMND        ;"END" PACKET FLAG IS "COMMAND"
                                ;------------------------------------------------- ----------------
                                ;END PACK SIZE:
000016                                  RSNDSZ == 14.           ;TOTAL BYTES IN COMMAND PACKET
                                ;MESSAGE PACK SIZE:
000012                                  RSMSIZ == 12            ;10. BYTES FOR BYTE COUNT INSIDE CMND PACK
                                ;DATA PACK SIZE:
000204                                  RSDASZ == 132.          ;TOTAL BYTES IN DATA PACKET
                                ;DATA + END PACK SIZE:
000222                                  RSDNSZ == RSDASZ+RSNDSZ
                                ;GET CHARACTERISTICS DATA PACKET SIZE
000034                                  RSGCDP == 28.           ;TOTAL BYTES FOR GET CHAR DATA PACKET
                                                                ;MINUS THE END PACKET
000016                                  RSSNSZ == RSMSIZ + 4            ;SIZE FOR SENDING COMMAND PACK
001036                                  RCBFSZ== 4*RSDASZ+RSNDSZ        ;4 DATA PAKS AND END PACK
                                                                        ;IS SIZE OF RCV BUFFERS
                                ;------------------- ---------- ------- - --- -- - -- -- - --
                                ;       THE OP CODES ARE:
000100                                  RSSEND == 100           ;END PACK DESCRIPTOR
000003                                  RSSWR  == 3             ;WRITE
000002                                  RSSRD == 2              ;READ
000005                                  RSSSEK == 5             ;SEEK
000012                                  RSSGET == 12            ;GET CHARACTERISTICS
000000                                  RSSNOP == 0             ;NO-OPERATION
000001                                  RSSNIT == 1             ;INITIALIZE
000007                                  RSSSLF == 7             ;SELF TEST
                                ;------------------- ---------------- ------         -- -     --
                                ;THE SUCCESS CODES ARE:
177720  ESABO    ==-48.          ;BAD COMMAND FROM HOST
177767  ESNCRT   ==-9.          ;NO CARTRIDGE
177770  ESNONX   ==-8.          ;NO DRIVE
000000  ESOK     ==0             ;OP COMPLETE SUCCESS
177776  ESPART   ==-2            ;PARTIAL OP
177740  ESSK     ==-32.          ;SEEK ERROR
000001  ESTRY    ==1             ;RETRYS OCCURRED
177765  ESWLOC   ==-11.          ;WRITE PROTECTED
177737  ESNOMO   ==-33.          ;MOTOR STOPPED
177720  ESCMD    ==-48.          ;COMMAND ERROR
177711  ESREC    ==-55.          ;BAD RECORD NUMBER.
177757  ESCKS    ==-17.          ;TU CHKSUM ERROR
177777  ESSLF    ==-1.           ;SELF TEST ERROR
177757  ESCKSM=ESCKS
177757  ESWR=ESCKS
177757  ESRD=ESCKS
                                ;-------------- ---- --   ------   ---- -- -    ----
```

```
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                          MSHDRD
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                          MSSELF
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          002324                              .EVEN
4106  002324  054523  052123  046505       MSNLOG:: .ASCIZ /SYSTEM ERROR/            ;DIAGNOSTIC HUNG. BETTER RE-BOOT
4107          002342                              .EVEN
4108  002342  040502  020104  040504       MSBDA::  .ASCIZ  /BAD DATA IN PACKET/     ;HOST DATA CHECK FOUND ERROR, DEVICE MAY
4109          002366                              .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          002516                              .EVEN
4118  002516  051127  052111  020105       MSWPRO:: .ASCIZ /WRITE PROTECTION/        ;CARTRIDGE WRITE PROTECT TAB MISSING OR
4119          002540                              .EVEN                               ;SWITCH BAD
4120  002540  042522  044503  053105       MSRNIT:: .ASCIZ /RECIEVING INIT/          ;DEVICE SENT INIT REQUEST
4121          002560                              .EVEN
4122  002560  047510  052123  043040       MSHCHK:: .ASCIZ /HOST FOUND PACKET CHECKSUM ERROR/  ;DEVICE SENT PACK WITH
4123          002622                              .EVEN                               ;BAD CHECKSUM
4124  002622  040503  023516  020124       MSNIT::  .ASCIZ  /CAN'T INIT/             ;DEVICE SENT BYTE OTHER THAN "CONTINUE"
4125          002636                              .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          002706                              .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 RECORD NUMBER
4133          002742                              .EVEN
4134  002742  051127  047117  020107       MSWRSP:: .ASCIZ /WRONG SUCCESS CODE/      ;HOST COULD NOT DECIPHER CODE IN END PAC
4135          002766                              .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       MSQRSP:: .ASCIZ \INDECIPHERABLE FLAG BYTE\  ;HOST COULD NOT UNDERSTAND 1ST BYTE
4139          003034                              .EVEN                               ;RESPONSE FROM TU AS PROPER PROTOCOL
4140  003034  044524  042515  047440       MSTOSN:: .ASCIZ /TIME OUT ON SEND/        ;DLV 'READY' NEVER WENT HIGH
4141          003056                              .EVEN
4142  003056  042522  047503  027126       MSSFRD:: .ASCIZ /RECOV. DATA CHECK ERR ON RD OP/  ;TU58 RESPONDED WITH "DATA-CHE
4143          003116                              .EVEN                               ;ERROR ON READ OP. ;HOST RETRY(S) SUCCES
4144  003116  042522  047503  027126       MSSFWR:: .ASCIZ /RECOV. DATA CHECK ERR ON WR OP/  ;SAME BUT WR OR WR VERIFY OPER
4145          003156                              .EVEN
4146  003156  047125  042522  047503       MSHDRD:: .ASCIZ /UNRECOV. DATA CHECK ERR ON RD OP/  ;TU58 RESPONDED WITH "DATA C
4147          003220                              .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          003262                              .EVEN
4150  003262  046104  020126  051105       MSOVRN:: .ASCIZ /DLV ERROR IN RECEIVE/    ;DLV ERROR (THE CONTENTS PRINTED OUT)
4151          003310                              .EVEN
```

```
4153                                    .SBTTL  MISC STORAGE AND EQUATES
4154
4155
4156  003310  000000                    SYSTAT:: .WORD                  ;SYSTEM STATUS WORD
4157                                                                    ;BIT7-BIT15 = 1ST BYTE OF PACK RCVD
4158                                                                    ;BIT05 = NOT USED
4159                                                                    ;BIT04 = NOT USED
4160                                                                    ;BIT03 = NOT USED
4161                                                                    ;BIT02 =RETRY BAD FLAG BYTE
4162                                                                    ;BIT01 = RETRY OCCURRING
4163                                                                    ;BIT00 = CHKSUM IS ODD (TEMP STORAGE)
4164
4165  003312  000000                    TAPLEN:: .WORD                  ;# RECORDS
4166  003314  000000                    DEVPTR:: .WORD                  ;->NEXT UNIT ADDRESS
4167  003316  000000                    RCFLG::  .WORD                  ;TEMP STORE FOR GTBYTE
4168  003320  000000                    RCBCNT:  .WORD                  ;TEMP STORAGE FOR GTBYTE
4169  003322  000004                    FTLNM:   .WORD 4                ;NUMBER OF TRIES BEFORE ABORT
4170  003324  000000                    DONE::   .WORD                  ;1 IF TEST ALGORITHM COMPLETE
4171  003326  000000                    IDPTR::  .WORD                  ;PTR IN TESTID MACRO
4172  003330  000000                    TSTTOP:  .WORD                  ;POINTER TO CURRENT TEST
4173  003332  000010                    MXRTRY:: .WORD  8.              ;NUMBER OF RETRIES, BEFORE "HARD" ERROR
4174  003334  000000                    BLKER::  .WORD                  ;RECORD # FOR ERROR REPORT
4175  003336  000000                    SECREC:: .WORD                  ;RECORD # TO START AFTER SWITCHING TRACKS
4176  003340  000000                    PRNSIZ:: .WORD                  ;SIZE OF PACK FOR "PRNPAK"
4177  003342  000000                    ALLGON:: .WORD                  ;SECONDARY DON'T PRINT STATISTICS FLAG.
4178  003344  000000                    TEST9::  .WORD                  ;***** FLAG THAT INDICATES IF IN TST 9
4179
4180
4181
4182                                    ;               TIME OUT CONSTANTS:
4183
4184  003346  177777                                    CSNRDY::  -1            ;TIME OUT ON SEND DELAY (DLV NOT READY)
4185  003350  000226                                    CSRCVB::  150.          ;45 SEC. MULTIPLIER  REWIND TIME
4186                                                                            ;CAN BE OPTIMISED PER CPU
```

```
                                .SBTTL  DATA BLOCK FORMAT
                                ;-- -- ---------------------------  ---- ------------------------------
                                ;R5   > TOP OF 1 OF THE 8 DATA BLOCKS (1 PER UNIT) DURING EXECUTION
                                ;@R5 IS THE STATUS WORD CONTAINING:
                                                                    ;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  = TU58 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 0DATA 1CMD
                                                                    ;BIT3  = RETRY FLAG BYTE ERR.(SEND COMMAND PACK)
                                                                    ;BIT0,1,2=UNIT NO.
000000                          STATUS  ==      0.                  ;DEVICE STATE
000002                          RETRY   ==      2.                  ;# OF RETRIES
000004                          ABNDX   ==      4.                  ;ERROR NUMBER FOR LOG
                                ;R0                                 ;STORAGE FOR REGISTERS USED IN TEST BODY
                                ;R1                                 ;STORED WITH SWAPOW
                                ;R2                                 ;RETRIEVED WITH SWAPIN
                                ;R3                                 ;
                                ;R4                                 ;
000020                          TSTPC   ==      16.                 ;POINTER TO NEXT EXECUTABLE TEST INST.
000022                          RCSR    ==      18.                 ;DLV RCV STATUS ADDRESS
000024                          RCDB    ==      20.                 ;DLV RCV DATA ADDRESS
000026                          XMSR    ==      22.                 ;DLV SND STATUS ADDRESS
000030                          XMDB    ==      24.                 ;DLV SND DATA ADDRESS
000032                          XSPKNM  ==      26.                 ;THE NUMBER OF PACKETS TO RECEIVE
000034                          XSFLG   ==      28.                 ;THE EXPECTED FLAG OF 1ST PACKET
000036                          XSCNT   ==      30.                 ;THE EXPECTED COUNT OF 1ST PACKET
                                                                    ;FOR MULTIPLE PACKET RECIEVES (MAX.4)
                                ;       BLKW    8.                  ;CONSECUTIVE XSFLGS AND XSCNTS
000060                          DR      ==      48.                 ;DR==0 OR 1; BIT8,9 DRIVE SELECTED BY OPERATOR
000062                          TRK     ==      50.                 ;COUNTER FOR TRACK NUMBER
000064                          REC     ==      52.                 ;RECORD (BLOCK #)

000066                          TMP     ==      54.                 ;TEST MACRO REGISTER
000070                          SNDCNT  ==      56.                 ;THE # OF BYTES FOR SENDING PACKET
000072                          PATTEN  ==      58.                 ;DATA PATTERN-LOWER BYTE USED
000074                          DLV     ==      60.                 ;CONTENTS OF RCDB ON DLV ERROR
000076                          SUCCS   ==      62.                 ;SUCCESS CODE OF LAST END PACKET
000100                          CMDSNT  ==      64.                 ;TYPE OF COMMAND CURRENT IN EVEN BYTE; BIT15==VE

000102                          RCVBUF  ==      66.                 ;POINTER TO 542. BYTE BUFFER (4 DATA PAKS + END
000104                          PKPTR   ==      68.                 ;POINTER TO TOP OF PACKET
000106                          XSPTR   ==      70.                 ;POINTER TO CURRENTLY USED XSFLG OR XSCNT
000110                          WRTN0   ==      72.                 ;THE # OF 512. BYTE BLOCKS WRITTEN DR0
000112                          WRTN1   ==      74.                 ;THE # OF 512. BYTE BLOCKS WRITTEN DR1
000114                          RDN0    ==      76.                 ;THE # OF 512. BYTE BLOCKS READ DR0
000116                          RDN1    ==      78.                 ;THE # OF 512. BYTE BLOCKS READ DR1
```

```
                                ;AND THE ERROR LOG...                   +-----------+
                                ;SPLIT INTO A BYTE PER DRIVE:           ! DR1 ! DR0 !
                                ;                                       +-----------+
                                ;
                                ; ----------  ------ ----------------------------------------------
                                ;OFFSET IN DATA BLOCK           ;ERROR TYPE     ;ERRCODE;MSG CODE;SUC. CODE
                                ;---- -- ------------------------------------------------------------
                                ;
000120                          LGOFST  ==      80.             ;**RESERVED**
000122                          SOFTR   ==      82.             ;SOFT READ      ;SFTRD  ;MSSFRD ;ESCKSM
000124                          SOFTW   ==      84.             ;SOFT WRITE     ;SFTWR  ;MSSFWR ;ESSKSM
                                ;       WORD                    ;RECIEVED INIT  ;RCINIT ;MSRNIT ;******
                                ;       WORD                    ;BAD FLAG BYTE  ;OTL    ;MSQRSP ;******
                                ;
                                ;THEN THOSE CODES WHICH HAVE N TRIES BEFORE ABORT
                                ;
000132                          T4TRY   ==      90.             ;DLV ERROR      ;OVRN   ;MSOVRN ;******
000134                          BDATA   ==      92.             ;BAD DATA       ;BDCOM  ;MSDATA ;******
000136                          HARDR   ==      94.             ;HARD READ      ;HRDRD  ;MSHDRD ;ESCKSM
000140                          HARDW   ==      96.             ;HARD WRITE     ;HRDWR  ;MSHDWR ;ESCKSM
                                ;       WORD                    ; CHKSM AT HOST ;BDCHK  ;MSHCHK ;******
                                ;       WORD                    ;SEEK ERROR TOTAL;SKERR ;MSSKER ;******
000146                          T1TRY   ==      102.            ;WRITE PROTECT  ;WRLOCK ;MSWPRO ;ESWLOC
                                ;       WORD                    ;NO MOTOR       ;NOMOT  ;MSNOMO ;ESNOMO
                                ;       WORD                    ;CANT INIT      ;CNINIT ;MSNIT  ;******
                                ;       WORD                    ;PARTIAL OP     ;PARTL  ;MSPART ;ESPART
                                ;       WORD                    ;NO UNIT        ;NOUNIT ;MSUNIT ;ESNONX
                                ;       WORD                    ;COMMAND ERROR  ;CMNDER ;MSCMD  ;ESCMD
                                ;       WORD                    ;BAD RECORD NO  ;RECERR ;MSREC  ;ESREC
                                ;       WORD                    ;SELF TEST ERROR;SLFER  ;MSSELF ;******
                                ;       WORD                    ;WRONG SUC.CODE ;SUCOTL ;MSWRSP ;******
                                ;       WORD                    ;NO RESPONSE    ;TORCVB ;MSNRSP ;******
                                ;       WORD                    ;**RESERVED**
                                ;       WORD                    ;NO CARTRIDGE   ;NOCART ;MSNOTP ;ESNCRT
                                ;       WORD                    ;TIME OUT  SEND;TOSNDB  ;MSTOSN ;******
                                ;
000202                          BLKEND  ==      130.            ;OFFSET OF END OF STATISTICS (RESERVED)
000204                          TUVECT  ==      132.            ;VECTOR ADDRESS
000206                          SAVCNT  ==       134.           ;BYTE COUNT SAVED DURING RETRY ON WRITE OPERATIO
000210                          MRSP    ==      136.            ;***** FLAG INDICATING MRSP
000212                          BLKSIZ  ==      138.            ;** RESERVED **
                                ;-------------- ------------------------------ --  - -- - ----    -- -
```

```
                                        .SBTTL  DEVICE DATA BLOCK ALLOCATION

                                        ;TABLE OF DEVICE DATA BLOCK ADDRESSES

003352  003372                          BLKTBL::        .WORD   DEV0
003354  003604                                          .WORD   DEV1
003356  004016                                          .WORD   DEV2
003360  004230                                          .WORD   DEV3
003362  004442                                          .WORD   DEV4
003364  004654                                          .WORD   DEV5
003366  005066                                          .WORD   DEV6
003370  005300                          LSTDEV::        .WORD   DEV7

                                        ;AND STORAGE FOR EACH:

003372  000212                          DEV0:           .BLKB   BLKSIZ
003604  000212                          DEV1:           .BLKB   BLKSIZ
004016  000212                          DEV2:           .BLKB   BLKSIZ
004230  000212                          DEV3:           .BLKB   BLKSIZ
004442  000212                          DEV4:           .BLKB   BLKSIZ
004654  000212                          DEV5:           .BLKB   BLKSIZ
005066  000212                          DEV6:           .BLKB   BLKSIZ
005300  000212                          DEV7:           .BLKB   BLKSIZ
```

```
4329                                            .SBTTL  GLOBAL TEXT SECTION
4330
4331                                            ;
4332                                            ; NAMES OF DEVICES SUPPORTED BY PROGRAM
4333                                            ;
4334  005512                                            DEVTYP  <TU58 CONTROLLER>
 (4)  005512                                                                    L$DVTYP::
 (3)  005512  052524  034065  041440                                                .ASCIZ  /TU58 CO
 (3)  005520  047117  051124  046117
 (3)  005526  042514  000122
 (2)                                                                            .EVEN
4335
4347
4348
4366
```

```
4375                                    .SBTTL  SYSTEM MACRO DEFINITIONS
4376
4377                                            .MACRO  PUSH    ,REG
4378
4379                                            .NLIST
4380                                            .LIST ME
4381                                            .LIST
4382                                                            MOV     REG,-(SP)
4383
4384
4385                                             NLIST
4386                                            .NLIST ME
4387                                            .LIST
4388                                            .ENDM
4389
4390                                            .MACRO  POP,REG
4391
4392                                            .NLIST
4393                                            .LIST ME
4394                                            .LIST
4395                                                            MOV     (SP)+,REG
4396
4397                                            .NLIST
4398                                            .NLIST ME
4399                                            .LIST
4400                                            .ENDM
4401
4402                                    ;++
4403                                    ;THE MACRO 'SWAPIN' RETRIEVES THE TEST REGISTERS WHICH WERE SAVED
4404                                    ;IN THE DEVICE DATA BLOCK.
4405                                    ;--
4406
4407                                            .MACRO  SWAPIN
4408
4409                                            .NLIST
4410                                            .LIST ME
4411                                            .LIST
4412                                                            MOV     6.(R5),R0
4413                                                            MOV     8.(R5),R1
4414                                                            MOV     10.(R5),R2
4415                                                            MOV     12.(R5),R3
4416                                                            MOV     14.(R5),R4
4417
4418                                            .NLIST
4419                                            .NLIST ME
4420                                            .LIST
4421                                            .ENDM
4422
4423                                    ;++
4424                                    ;THE MACRO 'SWAPOW' SAVES THE CURRENT STATE OF THE UNIT IN THE DRIVE
4425                                    ;DATA BLOCK IN SO THAT THE SCHEDULER MAY 'SWAPIN' ANOTHER UNIT.
4426                                    ;-
4427
4428                                            .MACRO  SWAPOW
4429
4430                                            .NLIST
```

```
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                                    ;THE WRITE MACRO IMPLEMENTS THE COMPLETE PROTOCOL NECESSARY TO BUILD
4447                                    ;A COMMAND PACKET AND SUBSEQUENT DATA PACKETS (UNTIL THE BYTE COUNT
4448                                    ;(BCNT) IS SATISFIED).
4449                                    ;
4450                                    ;SETS UP THE EXPECTED PROTOCOL RESPONSES:  THE NUMBER OF PACKETS
4451                                    ;(XSPKNM) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT).  CALLS
4452                                    ;'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
4453                                    ;CHECKSUM.
4454                                    ;
4455                                    ; INPUTS  - DEVICE BLOCK @R5
4456                                    ;           TRBUF - BUFFER ADDRESS
4457                                    ;           UNIT'S TEST REGISTERS FROM 'SWAPIN'
4458                                    ; OUTPUTS - SNDCNT(R5) = # OF BYTES TO SEND
4459                                    ;           XSPKNM = # OF PACKETS EXPECTED
4460                                    ;           XSFLG = FLAG BYTE OF 1ST PACKET
4461                                    ;           XSCNT = BYTE COUNT OF 1ST PACKET
4462                                    ;             . ***
4463                                    ;             .   *  SUBSEQUENT XSFLGS
4464                                    ;             .    >
4465                                    ;             .   *  AND XSCNTS
4466                                    ;             . ***
4467                                    ;--
4468
4469                                            .MACRO  TUWRIT  PTRN,REC,BCNT,DR,VER,?A,?B,?C,?D,?E,?F,?G,?H,?T
4470
4471                                            .NLIST
4472                                            .LIST ME
4473                                            .LIST
4474                                    T:      MOV     #TRBUF,R0       ;MAKE COMMAND PACKET:
4475                                            MOVB    #RSCMND,@R0     ;COMMAND FLAG
4476                                            MOVB    #RSMSIZ,1(R0)   ;THIS SIZE
4477                                            MOVB    #RSSWR,2(R0)    ;INSERT OP CODE-WRITE
4478                                            MOVB    VER,3.(R0)      ;VERIFY (1 OR 0)
4479                                            MOVB    DR,4.(R0)       ;DRIVE #
4480                                            MOVB    #020,5.(R0)     ;MAINTENANCE MODE SWITCH
4481                                            CLR     6.(R0)          ;NO SEQUENCE #
4482                                            MOV     BCNT,8.(R0)     ;TOTAL COUNT TO WRITE
4483                                            MOV     REC,10.(R0)     ;AT  RECORD N
4484                                            MOV     #RSMSIZ,R1      ;THE PACKET SIZE PLUS+2
4485                                            TST     (R1)+           ;(FLAG AND COUNT) INTO R
4486                                            MOV     #RSSNSZ,SNDCNT(R5)  ;LOAD THE  SIZE TO S
4487                                            CALL    CHKSUM          ;R0 --> R1=COUNT
4488                                            MOV     R1,(R0)         ;PUT CHKSUM IN PACKET
4489                                                                    ;SET UP EXPECTATIONS:
4490                                            MOV     #RSCONT,XSFLG(R5)   ;THE FLAG
4491                                            MOV     #1,XSCNT(R5)    ;THE COUNT
4492                                            MOV     #1,XSPKNM(R5)   ;THE # PACKETS EXPECTED
4493                                            MOV     BCNT,R2         ;GET # OF DATA BYTES
4494                                            CALL    RSVP            ;SEND (AND RETURN TO SCH
4495                                            BIT     #BIT3,@R5       ;FLAG BYTE ERROR?
4496                                            BNE     T               ;YES
4497                                            BIC     #BIT12,@R5      ;FLAG FOR LAST PACKET
4498                                    A:      MOV     #TRBUF,R0       ;POINT TO TOP OF BUFFER
4499                                            CMP     R2,#128.        ;START DATA PACKET(S)
4500                                            BHI     B               ;BCNT > 128.!
```

```
4501                                                    MOV     R2,R1           ;BCNT<128.
4502                                                    BIS     #BIT12,@R5      ;SO LAST PACKET NOW
4503                                                    BR      C               ;USE REMAINING COUNT
4504                                            B:      MOV     #128.,R1        ;USE 128. BYTES
4505                                            C:      MOVB    R1,1(R0)        ;COPY COUNT TO BUFFER
4506                                                    MOV     R1,R3           ;R3=COUNTER TO LOAD BUFF
4507                                                    MOVB    #RSDATA,@R0     ;FLAG FIRST
4508                                                    TST     (R0)+           ;SKIP COUNT
4509                                            D:      MOVB    PTRN,(R0)+      ;INSERT DATA
4510                                                    DEC     R3              ;MORE?
4511                                                    BHI     D               ;YES
4512                                                    MOV     #TRBUF,R0       ;-->TOP AGAIN
4513                                                    MOVB    1(R0),R1        ;GET COUNT
4514                                                    BIC     #177400,R1      ;ZERO SIGN EXTEND
4515                                                    MOV     R1,SNDCNT(R5)   ;HOW MANY TO SEND PLUS
4516                                                    ADD     #4,SNDCNT(R5)   ;FLAG,COUNT,CHKSUM
4517                                                    ADD     #2,R1           ;COMPENSATE FOR FLAG + C
4518                                                    CALL    CHKSUM          ;FOR CHECKSUM CALC.
4519                                                    MOVB    R1,(R0)+        ;CHKSUM INTO PACKET
4520                                                    SWAB    R1              ;EVEN ON AN ODD
4521                                                    MOVB    R1,(R0)+        ;BYTE BOUNDARY
4522                                                    BIT     #BIT12,@R5      ;LAST DATA PACKET?
4523                                                    BEQ     E               ;NO
4524                                                    MOV     #RSEND,XSFLG(R5)  ;YES-EXPECT 'END'
4525                                                    MOV     #RSNDSZ,XSCNT(R5)  ;OF THIS SIZE
4526                                                    MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
4527                                                    BR      F               ;SEND
4528                                            E:      MOV     #RSCONT,XSFLG(R5)  ;(NOT LAST), EXPECT
4529                                                    MOV     #1,XSCNT(R5)    ;AND 1 BYTE
4530                                                    MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
4531                                            F:      CALL    RSVP            ;SEND PACKET
4532                                                                            ;AND RETURN TO SCHEDULER
4533                                                    BIT     #BIT3,@R5       ;FLAG BYTE RETRY?
4534                                                    BNE     T               ;YES
4535                                                    BIT     #BIT10,@R5      ;RETRY DATA ERROR?
4536                                                    BNE     G               ;YES
4537                                                    SUB     #128.,R2        ;NO, MORE DATA TO SEND?
4538                                                    BHI     A               ;YES
4539                                                    BR      H               ;NO
4540                                            G:      TURTRY  REC,BCNT,DR     ;RETRY HERE
4541                                                    BIT     #BIT10!BIT3,@R5 ;RETRY AGAIN?
4542                                                    BNE     G               ;YES
4543                                            H:      NOP                     ;DONE
4544
4545                                    .NLIST
4546                                    .NLIST ME
4547                                    .LIST
4548                                    .ENDM
4549
```

```
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                    ;(XSPKNM) 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 @R5
4562                    ;           UNITS TEST REGISTERS FROM SWAPIN
4563                    ;           TRBUF - BUFFER ADDRESS
4564                    ;
4565                    ; OUTPUTS -
4566                    ;           XSPKNM = # 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                            .MACRO  TUSEEK  REC,DR,?A
4578
4579                            .NLIST
4580                            .LIST ME
4581                            .LIST
4582
4583                                    A:      MOV     @TRBUF,R0       ;-->(POINT TO) XMIT BUFF
4584                                            MOVB    @RSCMND,@R0     ;FORM COMMAND MESSAGE PA
4585                                            MOVB    @RSMSIZ,1(R0)   ;THIS BIG
4586                                            MOVB    @RSSSEK,2(R0)   ;OP CODE IS SEEK
4587                                            MOV     REC,10.(R0)     ;TO THIS RECORD
4588                                            MOVB    DR,4.(R0)       ;AND WHICH DRIVE
4589                                            CLRB    3.(R0)          ;NO MODIFIER
4590                                            CLRB    5.(R0)          ;NO SWITCHES
4591                                            CLR     6.(R0)          ;NO SEQUENCE #
4592                                            CLR     8.(R0)          ;NO BYTE COUNT
4593                                            MOV     @RSMSIZ,R1      ;GET COUNT
4594                                            TST     (R1)+           ;PLUS FLAG + BCNT
4595                                                                    ;FOR CHECKSUM CALC
4596                                            CALL    CHKSUM          ;R0-->TOP   R1=0 OF BYTE
4597                                            MOV     R1,(R0)         ;INSERT INTO PACKET
4598                                                                    ;SET UP EXPECTATIONS:
4599                                            MOV     @RSSNSZ,SNDCNT(R5)  ;HOW MANY TO SEND
4600                                            MOVB    @RSCMND,XSFLG(R5)  ;EXPECT END PACK
4601                                            MOV     @RSNDSZ,XSCNT(R5)  ;COUNT WITH THIS
4602                                            MOV     #1.,XSPKNM(R5)  ;EXPECT ONLY 1 PACKET
4603
4604                                            CALL    RSVP            ;SEND
4605                                                                    ;AND RETURN TO SCHEDULER
4606                                            BIT     @BIT3,@R5       ;RETRY (FLAG BYTE ERROR)
4607                                            BNE     A               ;YES
```

```
4608
4609                                                    .NLIST
4610                                                    .NLIST ME
4611                                                    .LIST
4612                                                    .ENDM
```

```
4615                    ;++
4616                    ;THE RETRY MACRO IMPLIMENTS 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 @R5
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 CMND PACK:
4646                            MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
4647                            MOVB    #RSMSIZ,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    @R5             ;REDUCED?
4653                            BPL     E               ;NO
4654                            INCB    3(R0)           ;YES-CHANGE THRESHOLD
4655                    E:      MOV     BCNT,8.(R0)     ;# BYTES DESIRED
4656                            MOVB    #020,5.(R0)     ;MAINTENANCE MODE
4657                            CLR     6.(R0)          ;NO SEQUENCE #
4658                            MOV     #RSMSIZ,R1      ;SIZE OF PACKET
4659                            TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
4660                            MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND
4661
4662                            CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
4663                            MOV     R1,(R0)         ;INSERT IN PACKET
4664
4665                            MOV     BCNT,R1         ;SET EXPECTATIONS:
4666                                                    ;CALC # OF DATA PACKETS
4667                            MOV     #XSFLG,R3       ;OFFSET OF FLAG
4668                            ADD     R5,R3           ;ABS. ADDR. OF XSFLG
4669                            CLR     R2              ;PRESET
4670                    A:      INC     R2              ;# PACKETS EXPECTED
```

```
                                                        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,XSPKNM(R5)   ;SAVE # PACKETS TO EXPEC
                                                        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 IMPLIMENTS 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 @R5
;           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
        MOV     #XSFLG,R3       ;GET OFFSET
        ADD     R5,R3           ;ABS. ADDR. OF XSFLG
        CLR     R2              ;PRESET AS NONE
A:      INC     R2              ;# PACKETS EXPECTED
        MOV     @RSDATA,(R3)+   ;LOAD XSFLG
        MOV     #132.,(R3)+     ;AND EXPECTED COUNT
        SUB     #128.,R1        ;NEG RESULT LAST TIME
```

```
4746                                                    BLOS    C               ;LAST TIME
4747                                                    BR      A               ;MORE TO DO
4748                                                C:  INC     R2              ;ADD ONE FOR END PACK
4749                                                    MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPEC
4750                                                    MOV     #RSEND,(R3)+    ;EXPECT AN END ALSO...
4751                                                    MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES
4752                                                    CALL    RSVP            ;SEND
4753                                                                            ;AND RETURN TO SCHEDULER
4754                                                D:  BIT     #BIT10!BIT3,@R5 ;RETRY?
4755                                                    BEQ     B               ;NO.
4756                                                    TURTRY  REC,BCNT,DR     ;YES
4757                                                    BR      D               ;ANOTHER RETRY?
4758                                                B:  NOP                     ;NO
4759
4760
4761                                    .NLIST
4762                                    .NLIST ME
4763                                    .LIST
4764                                    .ENDM
```

```
4767                                    ;++
4768                                    ;THE SELF TEST MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO
4769                                    ;INITIATE A 'DIAGNOSE' SEQUENCE.
4770                                    ;
4771                                    ;SETS UP THE EXPECTED PROTOCOL RESPONSES:  THE NUMBER OF PACKETS
4772                                    ;(XSPKNM) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT).  CALLS
4773                                    ;'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
4774                                    ;CHECKSUM.
4775                                    ;
4776                                    ; INPUTS  - DEVICE BLOCK @R5
4777                                    ;           TRBUF - BUFFER ADDRESS
4778                                    ;           UNITS REGISTERS TEST FROM SWAPIN
4779                                    ;
4780                                    ; OUTPUTS - SNDCNT(R5) = # OF BYTES TO SEND
4781                                    ;           XSPKNM = # OF PACKETS EXPECTED
4782                                    ;           XSFLG = FLAG BYTE OF 1ST PACKET
4783                                    ;           XSCNT = BYTE COUNT OF 1ST PACKET
4784                                    ;             . ***
4785                                    ;             .   *  SUBSEQUENT XSFLGS
4786                                    ;             .    >
4787                                    ;             .   *  AND XSCNTS
4788                                    ;             . ***
4789                                    ;--
4790
4791                                            .MACRO  TUSELF  ?A
4792
4793                                            .NLIST
4794                                            .LIST ME
4795                                            .LIST
4796
4797                                    A:      MOV     #TRBUF,R0       ;FORM COMMAND PACKET
4798                                            MOVB    #RSCMND,@R0     ;COMMAND FLAG
4799                                            MOVB    #RSMSIZ,1(R0)   ;SIZE OF MESSAGE
4800                                            MOVB    #RSSSLF,2(R0)   ;SELF TEST OPERATION
4801                                            CLRB    3(R0)           ;NO MODIFIER.
4802                                            CLR     4(R0)           ;NO DRIVE OR SWITCHES
4803                                            CLR     6(R0)           ;NO SEQUENCE NUMBER
4804                                            CLR     8.(R0)          ;NO BYTES
4805                                            CLR     10.(R0)         ;NO RECORD #
4806                                            MOV     #RSMSIZ,R1      ;GET SIZE
4807                                            TST     (R1)+           ;+2 FOR CHECKSUM
4808                                            MOV     #RSSNSZ,SNDCNT(R5)  ;SIZE TO SEND
4809                                            CALL     CHKSUM         ;FORM CHECKSUM
4810                                            MOV     R1,(R0)         ;INSERT INTO PACKET
4811                                            MOV     #RSEND,XSFLG(R5)  ;EXPECT END.
4812                                            MOV     #RSNDSZ,XSCNT(R5)  ;THIS BIG
4813                                            MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
4814                                                                    ;SEND
4815                                            CALL    RSVP            ;RETURN TO SCHEDULER
4816                                            BIT     #BIT3,@R5       ;RETRY?(BAD FLAG)
4817                                            BNE     A               ;YES
4818
4819                                            .NLIST
4820                                            .NLIST ME
4821                                            .LIST
4822                                            .ENDM
```

```
4825                                    ;++
4826                                    ;THE TEST ID MACRO INTERFACES THE SUPERVISOR'S TEST DISPATCH TO THE
4827                                    ;DIAGNOSTIC'S FORMAT BY IMPLEMENTING CALLS THAT:  1) INITIALIZE THE
4828                                    ;PC OF THE TEST CODE (TSTPC(R5)),  2) ASSIGN THE 1ST DRIVES,  3) RUN
4829                                    ;THE TEST,  4) SWITCH DRIVES AND REINITIALIZE, 5) RUN THE TEST AGAIN.
4830                                    ;--
4831
4832                                            .MACRO  TSTID   ADDRES,?A
4833
4834                                            .NLIST
4835                                            .LIST ME
4836                                            .LIST
4837                                                            MOV     ADDRES,TSTTOP   ;SAVE ADDR OF TEST
4838                                                            CALL    SETUP           ;INIT UNITS TSTPC
4839                                                            CALL    SETDR           ;GET 1ST DRVS.
4840                                                            CALL    RUN             ;DO TEST
4841                                                            CALL    SWAPDR          ;GET NEXT DRVS.
4842                                                            BCC     A               ;BR NO 2ND DRVS
4843                                                            CALL    SETUP           ;REINIT UNITS TSTPC
4844                                                            CALL    RUN             ;REPEAT TEST
4845                                                    A:                              ;DONE
4846                                            .NLIST
4847                                            .NLIST ME
4848                                            .LIST
4849                                            .ENDM
4850                                    ;------------------------------ - ------ -     --- --  --  -----
```

```
 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                                   ;               BLKTBL - 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  005532  005002                  SWAPDR:: CLR    R2              ;FOR # OF DRIVE 1'S.
 4931  005534  012737  003352  005650          MOV     #BLKTBL,SWPTR   ;TABLE ADDR. OF 1ST UNIT
 4932  005542  017705  000102          1$:     MOV     @SWPTR,R5       ;GET DATA BLOCK ADDR.
 4933  005546  032715  100000                  BIT     #BIT15,@R5      ;ABORTED?
 4934  005552  001013                          BNE     3$              ;YES
 4935  005554  032765  000001  000060          BIT     #BIT0,DR(R5)    ;DID DR. 0?
 4936  005562  001007                          BNE     3$              ;NO, DID DR.1 1ST PASS
 4937  005564  032765  001000  000060          BIT     #BIT9,DR(R5)    ;YES; 1 SELECTED?
 4938  005572  001403                          BEQ     3$              ;NO, ALL DONE
 4939  005574  105265  000060                  INCB    DR(R5)          ;YES, SWAP
 4940  005600  005202                          INC     R2              ;ONE MORE TO TEST
 4941  005602  023727  005650  003370  3$:     CMP     SWPTR,#LSTDEV   ;LAST DEVICE?
 4942  005610  103004                          BHIS    4$              ;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
```

```
                                        ;++
                                        ; 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
                                        ;--
005652  012737  003352  005726          SETDR:: MOV     #BLKTBL,SETPTR  ;TABLE OF ADDR. 1ST UNIT
005660  017705  000042                  1$:     MOV     @SETPTR,R5      ;GET DATA BLOCK ADDR.
005664  105065  000060                          CLRB    DR(R5)          ;PRESET AS DR0
005670  032765  000400  000060                   BIT     #BIT8,DR(R5)    ;DO DR0?
005676  001002                                   BNE     2$              ;YES
005700  105265  000060                          INCB    DR(R5)          ;NO-USE DRIVE 1
005704  023727  005726  003370          2$:     CMP     SETPTR,#LSTDEV  ;MORE UNITS
005712  103004                                   BHIS    3$              ;NO-EXIT
005714  062737  000002  005726                   ADD     #2,SETPTR       ;YES-GET TABLE ENTRY
005722  000756                                   BR      1$              ;CONFIGURE THAT UNIT
005724  000207                          3$:     RETURN
005726  000000                          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$:      MOV    @CLRPTR,R5      ;GET DATA BLOCK
4996  005742  004737  005770                     CALL   CLRBUF          ;CLEAR IT'S RECEIVE BUFFER
4997  005746  023727  006022  003370             CMP    CLRPTR,#LSTDEV  ;LAST DEV?
4998  005754  103004                             BHIS   2$              ;YES
4999  005756  062737  000002  006022             ADD    #2,CLRPTR       ;-->NEXT
5000  005764  000764                             BR     1$              ;CONTINUE
5001  005766  000207                    2$:      RETURN
```

```
5004                                    ;++
5005                                    ; CLRBUF - CLEARS 1 UNIT'S INPUT BUFFER.
5006                                    ; INPUTS:  RCVBUF(R5) IS BUFFER START
5007                                    ;          RCBFSZ - SIZE OF RECEIVE BUFFER IN BYTES
5008                                    ;          RCBFSZ IS SIZE OF BUFFER
5009                                    ; OUTPUTS:  CLEARED AREA.
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  000102                    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  000207                            RETURN                  ;EXIT
5022  006022  000000                    CLRPTR: .WORD
```

```
 5025                                   ;**
 5026                                   ; SETUP - CALLED WITHIN EACH TEST TO INSERT BEGINNING ADDRESS OF THE
 5027                                   ;         TEST INTO ALL UNITS TEST PC'S.
 5028                                   ; INPUTS:       TSTTOP LOADED WITH TEST ALGORITHMS STARTING ADDR.
 5029                                   ;               BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
 5030                                   ;               LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
 5031                                   ;
 5032                                   ; OUTPUTS:      TSTPC(R5) FOR ALL UNITS
 5033                                   ;               DONE - CLEARED
 5034                                   ;--
 5035
 5036  006024  005037  003324           SETUP:: CLR     DONE            ;NOT DONE YET
 5037  006030  012737  003352  003326           MOV     #BLKTBL,IDPTR   ;TABLE TOP ADDR
 5038  006036  017705  175264           1$:     MOV     @IDPTR,R5       ;DEVICE'S DATA BLOCK
 5039  006042  013765  003330  000020           MOV     TSTTOP,TSTPC(R5);INSERT PC FOR TOP OF TEST
 5040  006050  023727  003326  003370           CMP     IDPTR,#LSTDEV   ;ALL UNITS SET?
 5041  006056  103004                           BHIS    2$              ;YES
 5042  006060  062737  000002  003326           ADD     #2,IDPTR        ;NO,GET NEXT POINTER
 5043  006066  000763                           BR      1$              ;SET HIM UP
 5044  006070  000207           2$:     RETURN                          ;DONE
```

```
5047                                    ;**
5048                                    ; RUN - IMPLEMENTS THE CALLS TO SEND PACKETS, RECEIVE PACKETS, THEN
5049                                    ;       CHECK ANSWERS DURING TEST RUN TIME.
5050                                    ; INPUTS:  DONE
5051                                    ; OUTPUTS: NONE
5052                                    ;--
5053
5054  006072  004737  006122           RUN::   CALL    NXTST           ;MAKE AND SEND NEXT PACK TO ALL
5055                                                                    ;UNABORTED UNITS
5056  006076  005737  003324                   TST     DONE            ;COMPLETE?
5057  006102  001006                           BNE     2$              ;YES
5058  006104  004737  007172                   CALL    GETANS          ;NO,GET ALL RESPONSES
5059
5060  006110                                   BREAK                   ;SUPERVISOR CHECK
 (3)  006110  104422                                                                    TRAP    C$BRK
5061
5062  006112  004737  010616                   CALL    CHKANS          ;CHECK ALL RESPONSES
5063  006116  000765                           BR      RUN             ;CONTINUE TILL DONE
5064  006120  000207                   2$:     RETURN
```

```
                                                .SBTTL  NXTST / THE SCHEDULER
                                                ;++
                                                ; NXTST - DISPATCH EXECUTION USING EACH UN-ABORTED UNIT'S TEST PROGRAM
                                                ;         COUNTER, (TSTPC(R5)). (THE POINTER TO THE TEST CODE THAT COMPRISES
                                                ;         MAKING A PACKET AND SENDING IT. CHECKS FIRST FOR ANY UN-ABORTED UNIT
                                                ;         THAT IS RETRYING EITHER A DATA ERROR OR A 'INDECIPHERABLE FLAG BYTE'
                                                ;         ERROR, IN ORDER TO SERVICE ONLY THAT UNIT THIS PASS.  INITS
                                                ;        NON-RETRYING UNITS IF NECESSARY. IF NO RETRIES,DISPATCH ALL
                                                ;         UNITS IN ROUND ROBIN FASHION.
                                                ;
                                                ; INPUTS:      (IMPLIED) DATA BLOCKS.
                                                ;              BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
                                                ;              LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
                                                ;
                                                ; OUTPUTS:     ERRSF IF ALL UNITS ARE ABORTED.(TO NOTIFY APT)
                                                ;              SYSTAT IS UPDATED
                                                ;--
006122  000240                                  NXTST:: NOP                     ;
006124  012737  003352  003314                          MOV     @#BLKTBL,DEVPTR ;UNIT 0 TO START
006132  017705  175156                          1$:     MOV     @DEVPTR,R5      ;GET DATA BLOCK
006136  005715                                         TST     @R5             ;ABORTED?
006140  100504                                         BMI     2$              ; YES... CHECK NEXT UNIT
006142  032715  000010                          3$:     BIT     #BIT3,@R5       ;NO-RETRY 'BAD FLAG'?
006146  001040                                         BNE     5$              ;YES...(SEND BREAK;THEN CMD PACK)
006150  032715  020000                                 BIT     #BIT13,@R5      ;NO-RETRYING STILL (NO END PACK YET)?
006154  001426                                         BEQ     7$              ;NO...
006156  032715  000400                                 BIT     #BIT8,@R5       ;RETRYING A WRITE?
006162  001453                                         BEQ     4$              ;NO...
006164                                               SWAPIN                  ;YES-GET DEVICE REGESTERS
 (1)    006164  016500  000006                                                 MOV     6.(R5),R0
 (1)    006170  016501  000010                                                 MOV     8.(R5),R1
 (1)    006174  016502  000012                                                 MOV     10.(R5),R2
 (1)    006200  016503  000014                                                 MOV     12.(R5),R3
 (1)    006204  016504  000016                                                 MOV     14.(R5),R4
 (1)
006210  020265  000206                                 CMP     R2,SAVCNT(R5)   ;CURRENT COUNT = SAVED COUNT? (WHERE WE STARTED)
006214  001036                                         BNE     4$              ;NO...(CONTINUE SENDING DATA PACKS)
006216  042737  000004  003310                       BIC     #BIT2,SYSTAT    ;YES-CLEAR RETRY FLAGS
006224  042715  020000                               BIC     #BIT13,@R5      ;
006230  000450                                       BR      2$              ;CHECK NEXT UNIT.
006232  032715  002000                          7$:     BIT     #BIT10,@R5      ;NO-RETRY DATA ERROR?
006236  001445                                         BEQ     2$              ;NO...ON TO NEXT UNIT
006240  052737  000002  003310                       BIS     #BIT1,SYSTAT    ;SET RETRY STATUS TO 'DATA ERROR' TYPE
006246  000424                                       BR      6$              ;YES...
                                                
006250                                          5$:     SWAPIN                  ;GET DEVICE REGISTERS
 (1)    006250  016500  000006                                                 MOV     6.(R5),R0
 (1)    006254  016501  000010                                                 MOV     8.(R5),R1
 (1)    006260  016502  000012                                                 MOV     10.(R5),R2
 (1)    006264  016503  000014                                                 MOV     12.(R5),R3
 (1)    006270  016504  000016                                                 MOV     14.(R5),R4
 (1)
006274  010265  000206                                 MOV     R2,SAVCNT(R5)   ;SAVE THE BYTE COUNT (FOR WRITE OPERATION)
                                                                                ;TO MARK HOW MANY DATA PACKS TO SEND
```

```
GLOBAL AREAS    MACY11 30(1046)  25-JAN-84  08:33  PAGE 25-1
CZTUUF.P11    25-JAN-84 08:09          NXTST / THE SCHEDULER                                SEQ 0046

 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,@TSTPC(R5)   ;DO TEST FOR
 5117  006350  000477                              BR      NXTRET          ;THIS UNIT ONLY-EXIT
 5118  006352  023727  003314  003370      2$:     CMP     DEVPTR,#LSTDEV  ;TRY NEXT UNIT?
 5119  006360  103004                              BHIS    NXTST2          ;NO
 5120  006362  062737  000002  003314              ADD     #2.,DEVPTR      ;YES,->NEXT
 5121  006370  000660                              BR      1$              ;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     #BLKTBL,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$:     SWAPIN                  ;NO,GET DEVICE REGISTERS R0-R4 CONTAINING TEST P
  (1)  006440  016500  000006                                              MOV     6.(R5),R0
  (1)  006444  016501  000010                                              MOV     8.(R5),R1
  (1)  006450  016502  000012                                              MOV     10.(R5),R2
  (1)  006454  016503  000014                                              MOV     12.(R5),R3
  (1)  006460  016504  000016                                              MOV     14.(R5),R4
  (1)
 5137  006464  004775  000020                      JSR     PC,@TSTPC(R5)   ;INITIATE 1 PACKET TRANSMISSION AND RETURN
 5138  006470  005715                      4$:     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  022737  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$:    BREAK                   ;SUPERVISOR BREAK
  (3)  006540  104422                                                                      TRAP    C$BRK
 5149  006542  005237  003342                      INC     ALLGON          ;SET DON'T-PRINT STATISTICS FLAG
```

```
 5150  006546                                  DOCLN                   ;EXIT
  (3)  006546  104444                                                            TRAP    C$DCLN
 5151  006550  000207                          NXTRET: RETURN
 5152
 5153  006552  000000                          ABONM:  .WORD                   ;THE NUMBER OF ABORTED UNITS
 5154  006554  046101  020114  047125          NOMOR:  .ASCIZ  /ALL UNITS ABORTED!/
 5155          006600                                  .EVEN
```

```
                                        .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 DC NOT SEND XOFF AND
                                        ;        PROCEED NORMALLY.
                                        ; INPUTS:  (SP) CONTAINS UNITS PC TO SAVE SINCE RSVP WAS CALLED.  THE
                                        ;          NUMBER PACKETS EXPECTED (XSPKNM), 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.
                                        ;          XSPKNM(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 @R5 - 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 XSPKNM IS REPLICATED.
                                        ;           PACKET POINTER (PKPTR(R5)) POINTS TO TOP OF UNITS RECEIVE BUFFER
                                        ;           AREA (RCVBUF(R5)) FOR CURRENT UNIT.
                                        ;--
006600  000240                          RSVP::  NOP                     ;FINISH TEST
006602  012665  000020                        MOV     (SP)+,TSTPC(R5) ;SAVE WHERE YOU WERE IN TEST BODY AND
006606                                          SWAPOW                  ;SAVE TEST REGISTERS
006606  010065  000006                                                  MOV     R0,6.(R5)
006612  010165  000010                                                  MOV     R1,8.(R5)
006616  010265  000012                                                  MOV     R2,10.(R5)
006622  010365  000014                                                  MOV     R3,12.(R5)
006626  010465  000016                                                  MOV     R4,14.(R5)

                                                                        ;CORRECT FOR RETURN TO SCHEDULER
006632  022737  000002  003344                      CMP     #2,TEST9        ;***** IS THIS TEST 9
006640  001007                                      BNE     XFNSND          ;***** NO
006642  022765  000000  000210                      CMP     #0,MRSP(R5)     ;***** IF SO, IS THIS UNIT MODIFIED
006650  001523                                      BEQ     ENDRSP          ;***** YES
006652  012700  027746                  NOXOFF: MOV     #TRBUF,R0       ;FOR NORMAL PACKET SEND
006656  000404                                      BR      SND             ;SEND XOFF+PACKET
006660  012700  027745                  XFNSND: MOV     #TRBUF-1,R0     ;POINT TO XOFF
006664  005265  000070                          INC     SNDCNT(R5)      ;ONE MORE TO SEND, TOO.
006670  004737  007122                  SND:    CALL    SNDBYT          ;SEND BYTE
006674  005715                                  TST     @R5             ;R5--> TO STATUS BLK
006676  100510                                  BMI     6$              ;ABORTED? YES...QUIT
```

```
GLOBAL AREAS    MACY11 30(1046)  25-JAN 84  08:33  PAGE 25 4
CZTUUF.P11    25-JAN-84 08:09          RSVP / XOFF AND SEND A PACKET TO ALL DEVICES                    SEQ 0049

 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                  BISB    XSPKNM(R5),XSPKNM+1(R5)  ;REPLICATE LO. BYTE TO HI FOR GTPAKS, C
 5212  006726  005065  000076                          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 BUFFER 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,#RSCMND         ;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     RDN0(R5)            ;DRIVE 0
 5233  007036  000402                                  BR      4$                  ;
 5234  007040  005265  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     WRTN0(R5)           ;# BLKS WRITTEN DR0
 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
```

```
                                        .SBTTL  SNDBYT / OUTPUT A BYTE TO UNIT

                                        ;++
                                        ; SNDBYT - TEST 'READY' ON INTERFACE.  IF 'READY , SEND BYTE AND EXIT.
                                        ;          IF TIMED OUT, LOG ERROR.
                                        ; INPUTS - R0 = POINTER TO BUFFER
                                        ;        - IMPLIED UNIT DATA BLOCK
                                        ;        - CSNRDY - TIMEOUT CONSTANT
                                        ; OUTPUTS - R0 IS INCREMENTED.
                                        ; ERROR - NOT-READY-TO-SEND TIME OUT
                                        ;--

007122                                  SNDBYT:: PUSH   R1              ;ENTER R0- >BYTE
007122  010146                                   MOV    R1,-(SP)
(1)
(1)
007124  013701  003346                  4$:      MOV    CSNRDY,R1       ;GET TIMEOUT CONSTANT FOR NOT READY ERROR
007130  105775  000026                  1$:      TSTB   @XMSR(R5)       ;READY TO SEND?
007134  100412                                   BMI    2$              ;YES
007136                                           PUSH   R0              ;NO, SAVE R0
007136  010046                                   MOV    R0,-(SP)
(1)
(1)
007140                                           BREAK                  ;MONITOR BREAK
(3)     007140  104422                                                                  TRAP    C$BRK
007142                                           POP    R0              ;RESTORE
007142  012600                                   MOV    (SP)+,R0
(1)
007144  005301                                   DEC    R1              ;ABORTED?
007146  001370                                   BNE    1$              ;NO
007150  012704  000056                          MOV    #TOSNDB,R4      ;YES,SET CODE FOR TIMEOUT ERROR
007154  004737  012654                          CALL   LOG             ;LOG IT
007160  000402                                   BR     3$              ;QUIT
007162  112075  000030                  2$:      MOVB   (R0)+,@XMDB(R5) ;SEND IT
007166                                  3$:      POP    R1              ;RESTORE
007166  012601                                   MOV    (SP)+,R1
(1)
007170  000207                                   RETURN                 ;DONE
```

```
5284                                    .SBTTL  GETANS / GETS RESPONSES ROUND ROBIN USING "XON"
5285
5286                                    ;++
5287                                    ; GETANS - IF A UNIT IS RETRYING CLEAR HIS RECEIVE BUFFER (CLRBUF) AND GET
5288                                    ;          HIS RESPONSE (GTPKS1), ELSE, CLEAR ALL BUFFERS (CLRALL) AND
5289                                    ;          GET ALL RESPONSES (GTPKS8).
5290                                    ; INPUTS:  SYSTAT - SYSTEM STATUS WORD.
5291                                    ;
5292                                    ; OUTPUTS: SERVST = -1 IF NO RETRIES.
5293                                    ;--
5294
5295  007172  000240                    GETANS:: NOP                    ;1 UNIT IF RETRY;  ELSE ALL
5296  007174  032737  000006  003310            BIT     #BIT1!BIT2,SYSTAT       ;RETRY?
5297  007202  001010                            BNE     1$              ;YES
5298  007204  012737  177777  010362            MOV     #-1,SERVST      ;PRESET NO UNITS SERVICED
5299  007212  004737  005730                    CALL    CLRALL          ;CLEAR ALL INPUT BUFFERS
5300  007216  004737  007450                    CALL    GTPKS8          ;GET ALL REPLYS
5301  007222  000404                            BR      2$              ;EXIT
5302  007224  004737  005770            1$:     CALL    CLRBUF          ;RETRY-CLEAR 1 UNIT ONLY
5303                                                                    ;R5->UNIT BY NXTST
5304  007230  004737  007240                    CALL    GTPKS1          ;GET 1 REPLY
5305  007234  000207                    2$:     RETURN                  ;DONE
5306
5307  007236  000000                    GETPTR: .WORD
```

```
                                        .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
                                        ;          RSNDSZ - END PACKET SIZE
                                        ;
                                        ; OUTPUTS: SYSTAT UPPER BYTE = FLAG BYTE RECEIVED
                                        ;--
007240  000240                          GTPKS1:: NOP                    ;R5->THE UNIT
007242  012703  000034                          MOV     #XSFLG,R3       ;THE OFFSET VALUE OF FLAG
007246  060503                                  ADD     R5,R3           ;FORM THE ABSOLUTE ADDRESS
007250  010301                                  MOV     R3,R1           ;R3- >ADDR. OF EXPECTED FLAG
007252  062701  000002                          ADD     #2.,R1          ;R1-->ADDR. OF EXPECTED COUNT
007256  012700  007446                          MOV     #EXON,R0        ;R0=ADDRESS
007262  004737  007122                          CALL    SNDBYT          ;XON THE DEVICE
                                                                        ;*** TIME CRITICAL
007266  016500  000102                          MOV     RCVBUF(R5),R0   ;***--> TO THE BUFFER
007272  116502  000033                          MOVB    XSPKNM+1(R5),R2 ;***GET THE # OF PACKETS TO RECEIVE
007276  032702  177400                          BIT     #177400,R2      ;***SIGN UN-EXTEND
007302  011137  003320                  1$:     MOV     @R1,RCBCNT      ;***HOW MANY BYTES IT SHOULD BE
007306  011337  003316                          MOV     @R3,RCFLG       ;***WHAT THE FIRST BYTE SHOULD BE
007312  004737  010366                          CALL    GTBYTE          ;***GET THE ALL IMPORTANT FLAG
007316  032715  100000                          BIT     #BIT15,@R5      ;TIMEOUT?
007322  001050                                  BNE     4$              ;YES
007324  005300                                  DEC     R0              ;-> BYTE RECIEVED
007326  111037  003311                          MOVB    @R0,SYSTAT+1    ;SAVE IT AS FLAG BYTE
007332  121037  003316                          CMPB    @R0,RCFLG       ;1ST BYTE WHAT WAS EXPECTED?
007336  001420                                  BEQ     2$              ;YES
007340  121027  000002                          CMPB    @R0,#RSEND      ;NO, WAS IT END PAK?
007344  001006                                  BNE     14$             ;NO
007346  012737  000016  003320                  MOV     #RSNDSZ,RCBCNT  ;YES, USE END SIZE FOR COUNT
007354  012702  000001                          MOV     #1,R2           ;AND ASSUME IT'S LAST PACKET!
007360  000407                                  BR      2$              ;CONTINUE RECEIVE
007362  121027  000001                  14$:    CMPB    @R0,#RSDATA     ;WAS IT DATA?
007366  001026                                  BNE     4$              ;NO,CHKANS MAY FIND INIT...
007370  012737  000204  003320                  MOV     #RSDASZ,RCBCNT  ;YES, SET FOR DATA PAK SIZE
007376  005202                                  INC     R2              ;ONE MORE PACK THAN EXPECTED (END PACK)
                                        
007400  005200                          2$:     INC     R0              ;RESTORE TO -> NEXT BYTE
007402  005337  003320                  5$:     DEC     RCBCNT          ;THAT'S ONE LESS BYTE TO GO
007406  001411                                  BEQ     3$              ;DONE
007410  004737  010366                          CALL    GTBYTE          ;GET REST OF PACKET
007414  005765  000074                          TST     DLV(R5)         ;ERROR
007420  001011                                  BNE     4$              ;YES-ALL OVER
007422  032715  100000                          BIT     #BIT15,@R5      ;OR IF ABORTED
007426  001006                                  BNE     4$              ;THEN QUIT
007430  000764                                  BR      5$              ;CONTINUE RECEIVE
                                        
007432  005302                          3$:     DEC     R2              ;ONE LESS PACKET TO GO
007434  001403                                  BEQ     4$              ;MORE PACKETS IN TRANSACTION?
```

```
5366                                                                    ;YES
5367  007436  022121                        CMP     (R1)+,(R1)+       ;POINT TO NEW EXPECTED COUNT
5368  007440  022323                        CMP     (R3)+,(R3)+       ;AND FLAG,
5369  007442  000717                        BR      1$                ;AND RECEIVE,
5370  007444  000207                4$:     RETURN                    ;RETURN
5371
5372  007446     020                EXON:   .BYTE   RSXON
5373  007447     023                EXOFF:  .BYTE   RSXOFF
```

```
5376                                            .SBTTL  GTPKS8 / GET RESPONSES (NO RETRIES)
5377
5378                                            ;++
5379                                            ; GTPKS8 - IF IN TEST 9 AND THE UNIT IS NOT MODIFIED, SKIP THE REST
5380                                            ;          OF THE ROUTINE.  OTHERWISE;
5381                                            ;          SET ALL ABORTED UNITS SERVICED (SERVST: BIT POSITION).  UNTIL
5382                                            ;          ALL UNITS SERVICED (SERVST=0), IF NO MORE PACKETS, SET UNIT
5383                                            ;          SERVICED, ELSE, GET A FLAG BYTE FROM UNIT, DECREMENTING THE
5384                                            ;          NUMBER OF PACKETS LEFT.  CHECK TO SEE IF EXPECTED FLAG,
5385                                            ;          ADJUST COUNT IF NOT, GET REST OF PACKET.  IF WAS DATA PACK,
5386                                            ;          AND NOT IN TEST 9, SEND "XOFF" TO ENHANCE THROUGHPUT AND GO ON
5387                                            ;          TO NEXT UNIT (IF ANY).  IF IN TEST 9, DO NOT SEND "XOFFF".
5388                                            ; INPUTS:  (IMPLIED)UNITS DATA BLOCK POINTED TO BY R5.  NONE PASSED.
5389                                            ;          RSNDSZ - END PACK SIZE
5390                                            ;          RSDNSZ - DATA + END SIZE
5391                                            ;
5392                                            ; OUTPUTS: SYSTAT - UPPER BYTE=1ST BYTE RECEIVED, CURRENT UNIT
5393                                            ;--
5394
5395  007450  000240                            GTPKS8:: NOP                    ;GET ALL UNITS RESPONSES XOFF IF DATA PAK (THROU
5396  007452  022737  000002  003344                    CMP     #2,TEST9        ;***** IS THIS TEST 9
5397  007460  001006                                    BNE     1$              ;***** NO
5398  007462  022765  000000  000210                    CMP     #0,MRSP(R5)     ;***** IF SO, IS THIS UNIT MODIFIED
5399  007470  001002                                    BNE     1$              ;***** YES, CONTINUE NORMALLY
5400  007472  000137  010166                            JMP     ENDGP8          ;***** ELSE, SKIP ROUTINE
5401  007476  012737  003352  010364            1$:     MOV     #BLKTBL,GTPTR   ;->1ST
5402  007504  017705  000654            GTAGIN: MOV     @GTPTR,R5       ;GET DATA BLOCK
5403  007510  032715  100000                            BIT     #BIT15,@R5      ;ABORTED?
5404  007514  001404                                    BEQ     2$              ;NO
5405  007516  004737  010276                            CALL    SETSRV          ;YES-SET' SERVICED' AND
5406  007522  000137  010124                            JMP     GTDOWN          ;ON TO NEXT UNIT
5407  007526  105765  000033            2$:     TSTB    XSPKNM+1(R5)    ;NO, ANY PACKETS LEFT?
5408  007532  001004                                    BNE     3$              ;YES
5409  007534  004737  010276                            CALL    SETSRV          ;NO-HE'S DONE
5410  007540  000137  010124                            JMP     GTDOWN          ;SO ON TO NEXT UNIT
5411  007544  105365  000033            3$:     DECB    XSPKNM+1(R5)    ;NOW ITS ONE LESS PACKET
5412  007550  017537  000106  003316                    MOV     @XSPTR(R5),RCFLG  ;GET EXPECTED FLAG
5413  007556  062765  000002  000106                    ADD     #2,XSPTR(R5)    ;--> COUNT
5414  007564  017537  000106  003320                    MOV     @XSPTR(R5),RCBCNT  ;AND EXPECTED COUNT
5415  007572  022737  000002  003344                    CMP     #2,TEST9        ;***** IF TEST 9
5416  007600  001404                                    BEQ     1$              ;***** DO NOT SEND XON
5417  007602  012700  007446                            MOV     #EXON,R0        ;-> XON
5418                                                                            ;***TIME CRITICAL
5419  007606  004737  007122                            CALL    SMOBYT          ;***SEND IT
5420  007612  016500  000104            1$:     MOV     PKPTR(R5),R0    ;***->WHERE 1ST BYTE GOES
5421  007616  004737  010366                            CALL    GTBYTE          ;***GET IT
5422  007622  032715  100000                            BIT     #BIT15,@R5      ;ABORTED?
5423  007626  001404                                    BEQ     4$              ;NO-CONTINUE
5424  007630  105065  000033                            CLRB    XSPKNM+1(R5)    ;YES-NO MORE PACKETS EXPECTED
5425  007634  000137  010124                            JMP     GTDOWN          ;ON TO NEXT
5426  007640  005300                            4$:     DEC     R0              ;-->BYTE JUST RECEIVED
5427  007642  111037  003311                            MOVB    @R0,SYSTAT+1    ;SAVE IT
5428  007646  121037  003316                            CMPB    @R0,RCFLG       ;IS IT WHAT EXPECTED?
5429  007652  001436                                    BEQ     GTOK            ;YES
5430  007654  105065  000033            UNXPCT: CLRB    XSPKNM+1(R5)    ;NO, MUST BE LAST REPLY
5431  007660  121027  000002                            CMPB    @R0,#RSEND      ;MAYBE AN END PAK?
```

```
5432  007654  001004                        BNE     4$              ;NO
5433  007656  012737  000016  003320        MOV     #RSNDSZ,RCBCNT  ;YES, USE PROPER COUNT
5434  007674  000406                        BR      GTUM            ;AND GET IT
5435  007676  121027  000001          4$:   CMPB    @R0,#RSDATA     ;IS IT DATA?
5436  007702  001110                        BNE     GTDOWN          ;NO, ALL OVER, CHKANS WILL INIT UNIT
5437  007704  012737  000222  003320        MOV     #RSDNSZ,RCBCNT  ;YES, USE COUNT OF DATA + END PAK SURE TO FOLLOW
5438  007712  005200                  GTUM: INC     R0              ;WHERE TO STUFF THE REST
5439  007714  005337  003320          5$:   DEC     RCBCNT          ;ONE DOWN
5440  007720  001501                        BEQ     GTDOWN          ;NONE TO GO
5441  007722  004737  010366                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    XSPKNM+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 HANDSHAKING REQUIRED
5454  007762                                PUSH    R0              ;*** REV.- ELSE, TEST MRSP HANDSHAKE.
 (1)  007762  010046                                                MOV     R0,-(SP)
 (1)
 (1)
5455  007764  012737  000002  010274        MOV     #2,MRSDLY       ;*** REV.- DELAY FOR WAIT LOOP
5456                                        ;*** REV.- THIS IS THE BEGINNING DELAY LOOP
5457  007772  005000                  2$:   CLR     R0              ;*** REV.-
5458  007774  005300                  3$:   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    @RCSR(R5)       ;*** REV.- IF,   DONE SET,
5464  010012  001066                        BNE     ERRMOD          ;*** REV.- THEN, IT'S AN ERROR BECAUSE
5465                                        ;*** REV.-              THERE WAS NO MRSP HANDSHAKE.
5466  010014  012700  010272                MOV     #MODRSP,R0      ;*** REV.- ELSE, SEEMS TO BE OK, LETS
5467  010020  004737  007122                CALL    SNDBYT          ;*** REV.-       SEND A 'CONTINUE' AND
5468  010024                                POP     R0              ;*** REV.-       SEE IF HANDSHAKE WORKS.
 (1)  010024  012600                                                MOV     (SP)+,R0
 (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    XSPKNM+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$:    MOV     #EXOFF,R0       ;XOFF AND SEND TO
 5489  010120  004737  007122                  CALL    SNDBYT          ;ENHANCE THROUGHPUT
 5490  010124  062765  000002  000106  GTDOWN: ADD     #2.,XSPTR(R5)   ;NEXT XSFLG FOR NEXT TRY
 5491  010132  023727  010364  003370          CMP     GTPTR,#LSTDEV   ;DONE ONE CYCLE ALL UNITS?
 5492  010140  103005                          BHIS    1$              ;YES
 5493  010142  062737  000002  010364          ADD     #2,GTPTR        ;NEXT UNIT
 5494  010150  000137  007504                  JMP     GTAGIN          ;CONTINUE RECEIVE
 5495  010154  105737  010362          1$:     TSTB    SERVST          ;DONE SERVICING ALL PAKS
 5496                                                                  ;FROM ALL UNITS?
 5497  010160  001402                          BEQ     ENDGP8          ;YES
 5498  010162  000137  007450                  JMP     GTPKS8          ;NO, KEEP TRYING
 5499  010166  000207                  ENDGP8: RETURN                  ;RETURN
 5500
 5501  010170  000240                  ERRMOD: NOP                     ;*** REV.  MRSP ERROR
 5502  010172                                  PRINTF  #MESMRS,UNITNO
  (8)  010172  013746  027412                                                    MOV     UNITNO,
  (7)  010176  012746  010220                                                    MOV     #MESMRS,
  (6)  010202  012746  000002                                                    MOV     #2, (SP)
  (3)  010206  010600                                                            MOV     SP,R0
  (4)  010210  104417                                                            TRAP    C$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%/ERROR IN MRSP PROTOCOL!
 5506                                          .EVEN
 5507  010272     020                  MODRSP: .BYTE   RSCONT
 5508          010274                          .EVEN
 5509  010274  000000                  MRSDLY: .WORD
```

```
                                        .SBTTL  SETSRV / SET UNIT SERVICED

                                        ;++
                                        ; SETSRV - RESET THE BIT IN 'SERVST' CORRESPONDING TO THE UNIT NUMBER.
                                        ; INPUTS - SERVST - 'SERVICED' WORD
                                        ;        - @R5 = UNIT # (BITS 0, 1, 2)
                                        ; OUTPUTS - SERVST MODIFIED
                                        ;--

010276                                  SETSRV: PUSH    R5              ;SET UNIT SERVICED
(1) 010276  010546                              MOV     R5,-(SP)
(1)
(1)
010300                                          PUSH    R0
(1) 010300  010046                              MOV     R0,-(SP)
(1)
(1)
010302  011505                                  MOV     @R5,R5          ;GET STAT WD
010304  042705  177770                          BIC     #177770,R5      ;MASK UNIT #
010310  012700  010342                          MOV     #SRVTBL,R0      ;->TOP OF BIT TABLE
010314  005705                          1$:     TST     R5              ;RIGHT ONE?
010316  001404                                  BEQ     2$              ;YES
010320  062700  000002                          ADD     #2,R0           ;NO, ->NEXT
010324  005305                                  DEC     R5              ;1 LESS
010326  000772                                  BR      1$              ;CONTINUE
010330  041037  010362                  2$:     BIC     @R0,SERVST      ;MOW IT DOWN
010334                                          POP     R0
(1) 010334  012600                              MOV     (SP)+,R0
(1)
010336                                          POP     R5
(1) 010336  012605                              MOV     (SP)+,R5
(1)
010340  000207                                  RETURN                  ;RETURN

010342  000001                  SRVTBL: .WORD   BIT0            ;BIT POSITION LOOKUP TABLE
010344  000002                          .WORD   BIT1
010346  000004                          .WORD   BIT2
010350  000010                          .WORD   BIT3
010352  000020                          .WORD   BIT4
010354  000040                          .WORD   BIT5
010356  000100                          .WORD   BIT6
010360  000200                          .WORD   BIT7

010362  000000                  SERVST: .WORD
010364  000000                  GTPTR:  .WORD
```

```
                                        .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
                                        ;          O↑Æ 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
                                        ;--
010366  005037  010612                  GTBYTE:: CLR     GBTMP           ;TIMEOUT REGISTER
010372  013704  003350                           MOV     CSRCVB,R4       ;TIMEOUT ERROR CONSTANT (MULTIPLIER)
010376  105775  000022                  1$:      TSTB    @RCSR(R5)       ;READY?
010402  100013                                   BPL     3$              ;NO
010404  017565  000024  000074                   MOV     @RCDB(R5),DLV(R5)  ;GET ERROR + BYTE
010412  116520  000074                           MOVB    DLV(R5),(R0)+   ;COPY BYTE TO BUFFER
010416  005765  000074                           TST     DLV(R5)         ;ERROR?
010422  100472                                   BMI     4$              ;YES-EXIT
010424  005065  000074                           CLR     DLV(R5)         ;NO-RESET
010430  000467                                   BR      4$              ;AND EXIT
010432  005337  010612                  3$:      DEC     GBTMP           ;DEC T.O. CONSTANT
010436  001357                                   BNE     1$              ;STILL VALID

                                        ;CODE TO SEE ↑C DURING LONG SEEK OR REWIND
                                                                        ;HERE GBTMP=0
010440  010037  010614                           MOV     R0,GBTMP2       ;R0 MUST BE PRESERVED!
010444  012700  007447                           MOV     #EXOFF,R0       ;QUIET THE DEVICE
010450  004737  007122                           CALL    SNDBYT          ;BY SENDING XOFF
010454  105775  000022                  6$:      TSTB    @RCSR(R5)       ;CHARACTER SLOP OVER?
010460  100415                                   BMI     5$              ;YES
010462  005337  010612                           DEC     GBTMP           ;NO-WAIT A WHILE
010466  105737  010612                           TSTB    GBTMP           ;DONE WAITING?
010472  001370                                   BNE     6$              ;NO
010474                                           BREAK                   ;YES-NO SLOP OVER
 (3)    010474  104422                                                                   TRAP    C$BRK
010476  012700  007446                           MOV     #EXON,R0        ;START DEVICE TALKING
010502  004737  007122                           CALL    SNDBYT          ;AGAIN
010506  013700  010614                           MOV     GBTMP2,R0       ;RESTORE R0
010512  000426                                   BR      7$              ;END KLUGE
010514  013700  010614                  5$:      MOV     GBTMP2,R0       ;RESTORE R0
010520  017565  000024  000074                   MOV     @RCDB(R5),DLV(R5)  ;GET ERROR + BYTE
010526  116520  000074                           MOVB    DLV(R5),(R0)+   ;COPY BYTE TO BUFFER
010532  005765  000074                           TST     DLV(R5)         ;ERROR?
```

```
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     #TORCV3,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
```

```
                                        .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.
                                        ;--

010616  000240                  CHKANS:: NOP                     ;IF RETRY THEN CHECK ONE
                                                                 ;ELSE CHECK ALL
010620  032737  000006  003310          BIT     #BIT1!BIT2,SYSTAT       ;RETRYING?
010626  001403                          BEQ     CHK8             ;NO DO NORMAL
010630  004737  010726                  CALL    CHKPKS           ;YES DO SINGLE UNIT
                                                                 ;R5 -> UNIT
010634  000432                          BR      CHKANR           ;ALL DONE

010636  012737  003352  010724  CHK8:   MOV     #BLKTBL,CHKPTR   ;YOU KNOW ... TOP OF TABLE
010644  017705  000054          2$:     MOV     @CHKPTR,R5       ;GET UNIT'S BLOCK ADDRESS
010650  032715  100000                  BIT     #BIT15,@R5       ;ABORTED?
010654  001012                          BNE     3$               ;YES
010656  022737  000002  003344          CMP     #2,TEST9         ;***** IS THIS TEST 9
010664  001004                          BNE     1$               ;***** NO-CONTINUE NORMALLY
010666  022765  000000  000210          CMP     #0,MRSP(R5)      ;***** IF SO, IS THIS UNIT MODIFIED
010674  001402                          BEQ     3$               ;***** NO SKIP NEXT INSTR
010676  004737  010726          1$:     CALL    CHKPKS           ;NO, DO THIS GUY
010702  023727  010724  003370  3$:     CMP     CHKPTR,#LSTDEV   ;ALL DONE?
010710  103004                          BHIS    CHKANR           ;YES
010712  062737  000002  010724          ADD     #2,CHKPTR        ;NO,-->NEXT DEVICE
010720  000751                          BR      2$               ;DO DA

010722  000207                  CHKANR: RETURN

010724  000000                  CHKPTR: .WORD
```

```
                                        .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 TU58.  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
                                        ;--

010726  000240                          CHKPKS:: NOP                    ;CHECK WHAT WAS RECIEVED
010730  042715  000010                          BIC     #BIT3,@R5       ;CLEAR 'BAD FLAG' RETRY BIT
010734  016500  000102                          MOV     RCVBUF(R5),R0   ;GET BUFFER ADDR.
010740  116502  000032                          MOVB    XSPKNM(R5),R2   ;AND # OF PACKETS EXPECTED
010744  012703  000034                          MOV     #XSFLG,R3       ;THE OFFSET VALUE
010750  060503                              ADD     R5,R3           ;R3-->THIS UNIT XSFLG AGAIN
010752  010301                              MOV     R3,R1           ;COPY TO R1
010754  062701  000002                      ADD     #2,R1           ;R1-->XSBCNT FOR 1ST PACKET
010760  010065  000104                  1$: MOV     R0,PKPTR(R5)    ;POINT TO PACKET
010764  111037  003311                      MOVB    @R0,SYSTAT+1    ;SAVE RCV'D BYTE
010770  011137  003320                      MOV     @R1,RCBCNT      ;GET COUNT
010774  011337  003316                      MOV     @R3,RCFLG       ;AND FLAG
011000  121013                              CMPB    @R0,@R3         ;1ST BYTE=EXPECTED?
011002  001057                              BNE     5$              ;UH OH...
011004  121027  000020                      CMPB    @R0,#RSCONT     ;OK, IS IT 1 BYTE?
011010  001534                              BEQ     7$              ;YES...ONTO NEXT PACK
                                                                    ;NO, SO > 1 BYTE (NEVER EXPECT INIT!)
011012  013704  003320                      MOV     RCBCNT,R4       ;EXPECTED, SO COUNT MUST BE RIGHT
011016  005744                              TST     -(R4)           ;ADJUST FROM RECEIVE COUNT TO COUNT FOR CHECKSUM
011020  004737  013770                      CALL    CKCKSM          ;CHECK CHECKSUM
011024  103005                              BCC     2$              ;NO CARRY...NO INCORRECT
011026  012704  000022                      MOV     #BDCHK,R4       ;ERROR
011032  004737  012654                      CALL    LOG             ;LOG IT
011036  000521                              BR      7$              ;ON TO NEXT PACK
011040  122710  000002                  2$: CMPB    #RSEND,(R0)     ;END PAK?
011044  001005                              BNE     3$              ;NO
011046  004737  011322                      CALL    CHKEND          ;YES-CHECK
011052  012702  000001                      MOV     #1,R2           ;LAST PACKET
011056  000511                              BR      7$              ;AND FALL THROUGH
011060  122710  000001                  3$: CMPB    #RSDATA,@R0     ;DATA PAK?
011064  001012                              BNE     4$              ;NO
```

```
GLOBAL AREAS    MACY11 30(1046)  25-JAN-84  08:33  PAGE 33-1
CZTUUF.P11    25-JAN-84 08:09          CHKPKS / DECIPHERS RESPONSE OF UNIT POINTED TO BY R5 /                SEQ 0062

 5718  011066  022737  000001  003344                  CMP     #1,TEST9        ;***** IS THIS TEST 9
 5719  011074  001003                                  BNE     11$             ;***** NO-CONTTINUE NORMALLY
 5720  011076  005065  000210                          CLR     MRSP(R5)        ;***** CLR MRSP FLAG
 5721  011102  000402                                  BR      12$             ;***** SKIP INSTR
 5722  011104  004737  014570              11$:        CALL    COMPAR          ;YE'  HECK DATA
 5723  011110  000474                      12$:        BR      7$              ;ALL DONE?
 5724  011112  052715  020010              4$:         BIS     #BIT3!BIT13,@R5 ;SET 'BAD FLAG' RETRY FLAGS
 5725  011116  012704  000010                          MOV     #OTL,R4         ;OUT TO LUNCH
 5726  011122  005765  000074                          TST     DLV(R5)         ;AH,BUT DLV ERROR?
 5727  011126  001402                                  BEQ     20$             ;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$:         CMPB    #RSINIT,@R0     ;INIT?
 5735  011146  001007                                  BNE     6$              ;NO
 5736  011150  052715  020010                          BIS     #BIT3!BIT13,@R5 ;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$:         CMPB    #RSDATA,@R0     ;DATA PAK?
 5741  011172  001013                                  BNE     9$              ;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,R0      ;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
```

```
                                        .SBTTL  CHKEND / CHECK SUCCESS AND DETERMINE RETRY STATUS /

                                        ;++
                                        ; CHKEND - IF RETRYING;DETERMINE IF DATA ERROR OR BAD FLAG BYTE ERROR RETRY.
                                        ;
                                        ;       IF RETRYING BAD FLAG:RESET RETRY FLAG(SINCE OPERATION IS COMPLETE),
                                        ;       AND CHECK SUCCESS CODE.
                                        ;        IF RETRYING DATA ERROR ; CHECK SUCCESS CODE AND IF 0, PRINT RECOVERED,
                                        ;          SOFT ERROR, END RETRY STATUS.  IF NOT 0 AND WAS STILL "DATA
                                        ;          CHECK" ERROR - DETERMINE WHETHER TO CONTINUE ANOTHER RETRY OR
                                        ;          LOG "UNRECOVERABLE" ERROR.
                                        ;
                                        ;          IF NOT RETRYING DATA ERROR;  CHECK IF 'DATA CHECK' ERROR SUCCESS CODE
                                        ;          AND IF SO,START RETRY, ELSE EXIT.
                                        ; INPUTS:  IMPLIED  UNITS DATA BLOCK
                                        ; OUTPUTS: RETRY (SYSTAT BIT 1AND 2), (BIT10 @R5) RESET IF RETRYING.
                                        ;          -  DATA COMARE ERROR (BIT6 @R5) CLEARED.
                                        ;          -   REDUCED/NORMAL GAIN (BIT7 @R5) ADJUSTED
                                        ;--

011322  000240                          CHKEND:: NOP
011324                                          PUSH    R0              ;R0 --> END PAK
 (1)    011324  010046                                                  MOV     R0,-(SP)
 (1)
 (1)
011326                                          PUSH    R4
 (1)    011326  010446                                                  MOV     R4,-(SP)
 (1)
 (1)
011330  032737  000006  003310          1$:     BIT     #BIT1!BIT2,SYSTAT       ;RETRYING?
011336  001406                                  BEQ     NOREE           ;NO-CHECK NORMALLY
011340  032737  000004  003310                  BIT     #BIT2,SYSTAT    ;IS IT BAD FLAG TYPE?
011346  001454                                  BEQ     CHKREE          ;NO(DATA TYPE)
011350  042715  020000                          BIC     #BIT13,@R5      ;YES, SO IF END PACK THEN RETRY'S COMPLETE
011354  004737  012340                  NOREE:  CALL    CHKSUC          ;CHECK SUCCESS CODE
011360  032715  100000                          BIT     #BIT15,@R5      ;ABORTED?
011364  001402                                  BEQ     3$              ;NO,CONTINUE
011366  000137  012044                          JMP     CHKRET          ;YES,EXIT
011372  105765  000077                  3$:     TSTB    SUCCS+1(R5)     ;NO; HOW'D WE DO?
011376  001013                                  BNE     CHKERR          ;NOT SO GOOD.
011400  032715  000100                          BIT     #BIT6,@R5       ;OK, HOST FIND DATA PAK ERROR?
011404  001002                                  BNE     2$              ;YES
011406  000137  012044                          JMP     CHKRET          ;NO
011412  012704  000014                  2$:     MOV     #BDCOM,R4       ;YES;  JUST BAD DATA-NO DATACHK ERR
011416  004737  012654                          CALL    LOG             ;BAD DATA IN PACKET
011422  000137  012044                          JMP     CHKRET          ;QUIT
011426  032715  001000                  CHKERR: BIT     #BIT9,@R5       ;BAD SUCCESS; TU DATA CHK ERROR?
011432  001002                                  BNE     1$              ;YES
011434  000137  012044                          JMP     CHKRET          ;NO. ALL DONE.
011440  052715  002000                  1$:     BIS     #BIT10,@R5      ;YES-START RETRY
011444  012765  000001  000002                  MOV     #1,RETRY(R5)    ;CALL IT 1ST
011452                                          PRINTX  #RTRYN,RETRY(R5)  ;** PRINT **
 (8)    011452  016546  000002                                                                  MOV     RETRY(R5
 (7)    011456  012746  012224                                                                  MOV     #RTRYN,
 (6)    011462  012746  000002                                                                  MOV     #2,-(SP)
 (3)    011466  010600                                                                          MOV     SP,R0
```

```
GLOBAL AREAS    MACY11 30(1046)  25-JAN-84  08:33  PAGE 35-1
CZTUUF.P11    25-JAN-84 08:09          CHKEND / CHECK SUCCESS AND DETERMINE RETRY STATUS /                    SEQ 0065

  (4)  011470  104415                                                                    TRAP    C$PNTX
  (4)  011472  062706  000006                                                            ADD     #6,SP
5824   011476  000562                    BR      CHKRET          ;ALL DONE
5825   011500  004737  012340  CHKREE:   CALL    CHKSUC          ;CHECK SUCCESS CODE
5826   011504  105765  000077            TSTB    SUCCS+1(R5)     ; SUCCESSFUL YET?
5827   011510  001054                    BNE     UNSUC           ;NO, CHECK COUNT
5828   011512                            PRINTX  #RECOV,RETRY(R5)
  (8)  011512  016546  000002                                                            MOV     RETRY(R5
  (7)  011516  012746  012064                                                            MOV     #RECOV,
  (6)  011522  012746  000002                                                            MOV     #2,-(SP)
  (3)  011526  010600                                                                    MOV     SP,R0
  (4)  011530  104415                                                                    TRAP    C$PNTX
  (4)  011532  062706  000006                                                            ADD     #6,SP
5829   011536  105715                    TSTB    (R5)            ;DETERMINE THRESHOLD
5830   011540  100411                    BMI     2$              ;IT'S MODIFIED
5831   011542                            PRINTX  #THRSLO         ;NORMAL
  (7)  011542  012746  012144                                                            MOV     #THRSLO,
  (6)  011546  012746  000001                                                            MOV     #1,-(SP)
  (3)  011552  010600                                                                    MOV     SP,R0
  (4)  011554  104415                                                                    TRAP    C$PNTX
  (4)  011556  062706  000004                                                            ADD     #4,SP
5832   011562  000410                    BR      3$
5833   011564                    2$:     PRINTX  #THRSHI         ;ENHANCED
  (7)  011564  012746  012172                                                            MOV     #THRSHI,
  (6)  011570  012746  000001                                                            MOV     #1,-(SP)
  (3)  011574  010600                                                                    MOV     SP,R0
  (4)  011576  104415                                                                    TRAP    C$PNTX
  (4)  011600  062706  000004                                                            ADD     #4,SP
5834   011604  032715  000400    3$:     BIT     #BIT8,@R5       ;WRITE OR READ OPERATION?
5835   011610  001003                    BNE     4$              ;WRITE
5836   011612  012704  000002            MOV     #SFTRD,R4       ;READ
5837   011616  000402                    BR      5$
5838   011620  012704  000004    4$:     MOV     #SFTWR,R4       ;WRITE
5839   011624  004737  012654    5$:     CALL    LOG
5840   011630  005065  000002            CLR     RETRY(R5)       ;RESTORE TO NORMAL STATE
5841   011634  042715  002200            BIC     #BIT10!BIT7,@R5 ;NO RETRY, NORM THRESHOLD
5842   011640  000501                    BR      CHKRET          ;QUIT
5843
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  104414                                                                    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,-
```

```
 (6)  011730  012746  000002                                                          MOV     #2,-(SP)
 (3)  011734  010600                                                                  MOV     SP,R0
 (4)  011736  104415                                                                  TRAP    C$PNTX
 (4)  011740  062706  000006                                                          ADD     #6,SP
5855  011744  032715  000200                  BIT     #BIT7,@R5       ;WAS NORMAL THRESHOLD?
5856  011750  001403                          BEQ     1$              ;YES-REDUCE GAIN
5857  011752  042715  000200                  BIC     #BIT7,@R5       ;NO-NORM
5858  011756  000432                          BR      CHKRET
5859  011760  052715  000200          1$:     BIS     #BIT7,@R5       ;REDUCED
5860  011764  000427                          BR      CHKRET          ;DONE
5861  011766  000240          HRD1:   NOP                             ;HERE IS HARD ERROR!
5862  011770                          PRINTX  #UNREC
 (7)  011770  012746  012244                                                          MOV     #UNREC,
 (6)  011774  012746  000001                                                          MOV     #1,(SP)
 (3)  012000  010600                                                                  MOV     SP,R0
 (4)  012002  104415                                                                  TRAP    C$PNTX
 (4)  012004  062706  000004                                                          ADD     #4,SP
5863  012010  032715  000400                  BIT     #BIT8,@R5       ;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,@R5 ;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,@R5       ;NO MORE HOST DATA CHECK ERROR
5874  012056                                  POP     R4
 (1)  012056  012604                                                  MOV     (SP)+,R4
 (1)
5875  012060                                  POP     R0
 (1)  012060  012600                                                  MOV     (SP)+,R0
 (1)
5876  012062  000207                          RETURN
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  012172  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  RETERR: .ASCIZ  /%AOTHER ERROR DURING RETRY : EXIT RETRY%N/
5890                                          .EVEN
```

```
                                                .SBTTL  CHKSUC / INTERPRET SUCCESS CODE /

                                                ;++
                                                ; CHKSUC - COPY SUCCESS CODE (BYTE) TO SUCCS+1(R5).  INTERPRET SUCCESS
                                                ;          AND IF NOT 0, LOG APPROPRIATE ERROR.
                                                ; INPUTS:  R0 POINTS TO END PACKET
                                                ;          @R5 - UNIT STATUS WORD
                                                ;          CMDSNT(R5) - COMMAND BYTE
                                                ;
                                                ; OUTPUTS: R4 IS ERROR NUMBER IF ERROR.
                                                ;            SUCCS(R5) UPDATED.
                                                ;            BIT9 @R5 SET ON DATA CHECK SUCCESS CODE
                                                ; -

012340  000240                                  CHKSUC:: NOP
012342  016065  000002  000076                          MOV     2(R0),SUCCS(R5) ;R0-->END PACKET
012350  122760  000000  000003                          CMPB    @ESOK,3(R0)     ;GET SUCCESS BYTE
012356  001535                                         BEQ     12$             ;COMPLETE SUCCESS-EXIT

012360  122760  000001  000003                          CMPB    @ESTRY,3(R0)    ;OK BUT RETRIES?
012366  001012                                         BNE     20$             ;NO
012370  126527  000100  000002                          CMPB    CMDSNT(R5),@RSSRD;A READ?
012376  001001                                         BNE     22$             ;NO

012400  000520                                         BR      10$             ;NO RETRIES IN MAINTENANCE!
012402  126527  000100  000003          22$:    CMPB    CMDSNT(R5),@RSSWR       ;A WRITE?
012410  001001                                         BNE     20$             ;NO
012412  000513                                         BR      10$             ;LOG IT
012414  122760  177737  000003          20$:    CMPB    @ESNOMO,3(R0)   ;NO MOTOR?
012422  001003                                         BNE     1$              ;NO
012424  012704  000030                                 MOV     @NOMOT,R4       ;YES-
012430  000506                                         BR      11$             ;LOG

012432  122760  177757  000003          1$:     CMPB    @ESCKS,3(R0)    ;"DATA CHECK" ERROR?
012440  001003                                         BNE     2$              ;NO
012442  052715  001000                                 BIS     @BIT9,@R5       ;SET DATA-CHK-ERROR FLAG
012446  000501                                         BR      12$             ;DONT LOG

012450  126527  000100  000007          2$:     CMPB    CMDSNT(R5),@RSSSLF  ;SELF TEST?
012456  001006                                         BNE     3$              ;NOPE
012460  105760  000003                                 TSTB    3(R0)           ;YES, NEG. IF ERROR
012464  100072                                         BPL     12$             ;OK

012466  012704  000044                                 MOV     @SLFER,R4       ;YES-ERROR
012472  000465                                         BR      11$             ;LOG IT

012474  122760  177740  000003          3$:     CMPB    @ESSK,3(R0)     ;SEEK ERROR?
012502  001005                                         BNE     4$              ;NO
012504  012704  000024                                 MOV     @SKERR,R4       ;YES-
012510  052715  040000                                 BIS     @BIT14,@R5      ;SET 'DOBRK' FLAG  *** REV E *** MISSING @
012514  000454                                         BR      11$             ;LOG

012516  122760  177767  000003          4$:     CMPB    @ESNCRT,3(R0)   ;NO CART?
012524  001003                                         BNE     5$              ;NO
012526  012704  000054                                 MOV     @NCART,R4       ;YES-
012532  000445                                         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     #CMNDER,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     #NOUNIT,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
```

```
                                        .SBTTL  LOG / TO LOG ERROR IN CORRECT PLACE

                                        ;**
                                        ; LOG - DETERMINE IF ERROR IS FATAL, NON-FATAL OR FATAL AFTER N TRIES
                                        ;       BY INDEX (ERROR #) INTO DEVICE DATA BLOCK.  ADD THE DRIVE # TO
                                        ;       INDICATE UPPER OR LOWER BYTE AND INCREMENT THAT ERROR UNLESS
                                        ;       THAT BYTE WOULD OVERFLOW.  DETERMINE IF EVL FLAG SET,AND IF SO,
                                        ;       CHECK THRESHOLD (EVLTHR) AND  PRINT APPROPRIATE ERROR MESSAGE
                                        ;       DESCRIPTION.  ABORT THE UNIT IF INDICATED THROUGH DODROP CODE.
                                        ; INPUTS:  R4 = ERROR CODE
                                        ; OUTPUTS: ABNDX(R5) = ERROR CODE.
                                        ;           DLV(R5) = 0
                                        ;         L$LUN = UNIT NUMBER
                                        ;--

012654                                  LOG::   PUSH    R0
012654  010046                                          MOV     R0,-(SP)
012656                                          PUSH    R1
012656  010146                                          MOV     R1,-(SP)
012660                                          PUSH    R3
012660  010346                                          MOV     R3,-(SP)
012662                                          PUSH    R4
012662  010446                                          MOV     R4,-(SP)

012664  011537  002074                          MOV     @R5,L$LUN       ;GET UNIT NUMBER
012670  042737  177770  002074                  BIC     #177770,L$LUN   ;MASK IT OFF
012676  010465  000004                          MOV     R4,ABNDX(R5)    ;SAVE INDEX IN CASE OF ABORT MESSAGE
012702  012703  000120                          MOV     #LGOFST,R3      ;OFFSET TO LOW ORDER BYTE (DRIVE0)
012706  060403                                  ADD     R4,R3           ;FORM INDEX OF PARAM. TO UPDATE
012710  060503                                  ADD     R5,R3           ;FORM ABSOLUTE ADDR. THIS UNIT
012712  004737  013660                          CALL    WHCHDR          ;SEE WHICH DRIVE T'WAS
012716  103001                                  BCC     2$              ;WAS DRIVE 0
012720  005203                                  INC     R3              ;DRIVE 1; POINT TO UPPER BYTE
012722  122713  000377                  2$:     CMPB    #255.,@R3       ;POTENTIAL OVERFLOW POSSIBLE?
012726  001005                                  BNE     LOGOK           ;NO
012730                                  LOGO:   ERRDF   0.,OVRFLO,ERRDES  ;YES
012730  104455                                                                  TRAP    C$ERDF
012732  000000                                                                  .WORD   0
012734  013554                                                                  .WORD   OVRFLO
012736  013210                                                                  .WORD   ERRDES
012740  000512                                  BR      ABO             ;ABORT UNIT
012742  105213                          LOGOK:  INCB    @R3             ;INCREMENT THE ERROR
012744  111304                                  MOVB    @R3,R4          ;TEMP'LY SAVE IT
012746  016503  000004                          MOV     ABNDX(R5),R3    ;GET INDEX AGAIN
012752  012701  002230                          MOV     #RSNTAB,R1      ;FORM ADRS OF MSG
012756  066501  000004                          ADD     ABNDX(R5),R1    ;LIKE THIS
012762  042701  000001                          BIC     #BIT0,R1        ;INSURE WORD BOUNDARY
012766  032737  000004  016774                  BIT     #EVL,FLGLOC     ;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     @R1,DFTL1+4     ;AND MESSAGE ADDR
 6025  013014                        DFTL1:  ERRDF   0,DFTL1,ERRDES  ;ERROR
  (4)  013014  104455                                                            TRAP    C$ERDF
  (5)  013016  000000                                                            .WORD   0
  (5)  013020  013014                                                            .WORD   DFTL1
  (5)  013022  013210                                                            .WORD   ERRDES
 6026  013024  000460                        BR      ABO             ;DROP IT
 6027  013026  120327  000014        LOGOK2: CMPB    R3,#BDCOM       ;'NEVER FATAL' TYPE?
 6028  013032  103011                        BHIS    NTSFT           ;NO
 6029  013034  010337  013046                MOV     R3,LOG1+2       ;YES, ERROR CODE
 6030  013040  011137  013050                MOV     @R1,LOG1+4      ;DESCRIPTION
 6031  013044                        LOG1:   ERRSOFT 0.,LOG1,ERRDES
  (4)  013044  104457                                                            TRAP    C$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     @R1,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                MOV     @R1,LOG3+4      ;DESCRIPTION
 6049  013134                        LOG3:   ERRSOFT 0,LOG3,ERRDES
  (4)  013134  104457                                                            TRAP    C$ERSOFT
  (5)  013136  000000                                                            .WORD   0
  (5)  013140  013134                                                            .WORD   LOG3
  (5)  013142  013210                                                            .WORD   ERRDES
 6050  013144  000414                        BR      LOGO            ;EXIT
 6051  013146  010337  013160        HRD:    MOV     R3,LOG3B+2      ;HARD ERROR CODE
 6052  013152  011137  013162                MOV     @R1,LOG3B+4     ;DESCRIPTION
 6053  013156                        LOG3B:  ERRHRD  0,LOG3B,ERRDES
  (4)  013156  104456                                                            TRAP    C$ERHRD
  (5)  013160  000000                                                            .WORD   0
  (5)  013162  013156                                                            .WORD   LOG3B
  (5)  013164  013210                                                            .WORD   ERRDES
 6054
 6055  013166  011500                ABO:    MOV     @R5,R0          ;GET UNIT NUMBER
```

```
 6056  013170  042700  177770                    BIC     #177770,R0      ;UN-SIGN EXTEND
 6057  013174                                    DODU    R0              ;USE LOGICAL # TO DROP
  (3)  013174  104451                                                                    TRAP    C$DODU
 6058  013176                          LOGO:     POP     R4              ;RESTORE
  (1)  013176  012604                                                    MOV     (SP)+,R4
  (1)
 6059  013200                                    POP     R3
  (1)  013200  012603                                                    MOV     (SP)+,R3
  (1)
 6060  013202                                    POP     R1
  (1)  013202  012601                                                    MOV     (SP)+,R1
  (1)
 6061  013204                                    POP     R0
  (1)  013204  012600                                                    MOV     (SP)+,R0
  (1)
 6062  013206  000207                          RETURN                    ;RETURN
```

```
                                        ;++
                                        ; ERRDES - CONTAINS CODE FOR EXTENDED ERROR INFORMATION:  DRIVE #,
                                        ;          BLOCK #, ETC.
                                        ; -
                                        
                                        ERRDES: BGNMSG  ERRDES          ;ERROR DESCRIPTION
                                        
                                                PUSH    R0
                                                                        MOV     R0,-(SP)
                                        
                                                PUSH    R2
                                                                        MOV     R2,-(SP)
                                        
                                                CLR     R2              ;PRESET TO DATA TYPE
                                                BIT     #BIT4,@R5       ;WHAT PACK TYPE?
                                                BEQ     2$              ;DATA
                                                INC     R2              ;COMMAND
                                        2$:     PRINTB  #UNIT,<B,DR(R5)>,R2,<B,SYSTAT+1>
                                                                                        CLR     -(SP)
                                                                                        BISB    SYSTAT+1
                                                                                        MOV     R2,-(SP)
                                                                                        CLR     -(SP)
                                                                                        BISB    DR(R5),(
                                                                                        MOV     #UNIT,-(
                                                                                        MOV     #4, (SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTB
                                                                                        ADD     #12,SP
                                                MOV     REC(R5),R0      ;RECORD NUMBER
                                                MOV     PATTEN(R5),R2   ;DATA EXPECTED
                                                PRINTB  #RECID,R0,<B,CMDSNT(R5)>,<B,R2>,<B,SUCCS+1(R5)>
                                                                                        CLR     -(SP)
                                                                                        BISB    SUCCS+1(
                                                                                        CLR     -(SP)
                                                                                        BISB    R2,(SP)
                                                                                        CLR     -(SP)
                                                                                        BISB    CMDSNT(R
                                                                                        MOV     R0,-(SP)
                                                                                        MOV     #RECID,
                                                                                        MOV     #5,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTB
                                                                                        ADD     #14,SP
                                                TST     DLV(R5)         ;DLV ERROR?
                                                BEQ     3$              ;NO
                                                PRINTB  #RECID2,DLV(R5) ;YES-PRINT
                                                                                        MOV     DLV(R5),
                                                                                        MOV     #RECID2,
                                                                                        MOV     #2,-(SP)
                                                                                        MOV     SP,R0
                                                                                        TRAP    C$PNTB
                                                                                        ADD     #6,SP
                                                CLR     DLV(R5)         ;RESET
                                        3$:     POP     R2              ;RESTORE
```

```
GLOBAL AREAS    MACY11 30(1046)  25-JAN 84  08:33  PAGE 38-1
CZTUUF.P11    25-JAN-84 08:09          LOG / TO LOG ERROR IN CORRECT PLACE                                    SEQ 0073

   (1)  013374  012602                                                          MOV     (SP)+,R2
   (1)
 6086   013376                                        POF     R0
   (1)  013376  012600                                                          MOV     (SP)+,R0
   (1)
 6087   013400                                        ENDMSG                  ;EXIT
   (3)  013400                                                                                  L10003:
   (3)  013400  104423                                                                                  TRAP    C$MSG
 6088   013402  040445  051104  053111        UNIT::  .ASCIZ  /%ADRIVE#  %01%A  PAK SENT %01%A FLAG RCVD %03%N/
 6089                                                  .EVEN
 6090   013462  040445  046102  041517        RECID:: .ASCIZ  /%ABLOCK#  %04%A COMMAND %02%A EXPCTD %03%A SUCCESS %03%N/
 6091           013554                                 .EVEN
 6092   013554  040503  023516  020124        OVRFLO: .ASCIZ  /CAN'T UPDATE ERROR OR STATISTIC:OVERFLOW PENDING/
 6093           013636                                 .EVEN
 6094   013636  040445  051040  042103        RECID2: .ASCIZ  /%A RCDB WAS %06%N/
 6095                                                  .EVEN
```

```
                                        .SBTTL  WHCHDR / SEE WHICH DRIVE IS ACTIVE

                                        ;++
                                        ; INPUTS:  DR(R5)
                                        ; OUTPUTS:  CARRY=DRIVE (1 OR 0)
                                        ;--

013660  000241                  WHCHDR:: CLC                     ;CLEAR CARRY

013662  105765  000060                    TSTB    DR(R5)          ;DR 0?
013666  001401                          BEQ     2$              ;YES
013670  000261                          SEC                     ;NO

013672  000207                  2$:     RETURN                  ;RETURN
```

```
                                .SBTTL  CHKSUM / FORM THE PACKET CHECKSUM

                                ;++
                                ; THE CHECKSUM IS A 16 BIT CHECKSUM WITH END-AROUND CARRY.
                                ;
                                ; INPUTS:  R0 -> (POINTS TO) TOP OF PACKET
                                ;          R1 = # OF BYTES
                                ; OUTPUTS:  R0 -> WHERE TO PUT CHECKSUM
                                ;           R1 = CHECKSUM
                                ;--

013674                          CHKSUM:: PUSH   R3
013674  010346                                   MOV     R3,-(SP)
013676                                  PUSH    R2
013676  010246                                   MOV     R2,-(SP)
013700  042737  000001  003310          BIC     #BIT0,SYSTAT    ;"CHECKSUM IS ODD" BIT
013706  032701  000001                  BIT     #BIT0,R1        ;AN ODD # OF BYTES?
013712  001403                          BEQ     1$              ;NO
013714  052737  000001  003310          BIS     #BIT0,SYSTAT    ;YES

013722  006001                  1$:     ROR     R1              ;/2 FOR WORDS

013724  005003                  2$:     CLR     R3              ;PREP CHECKSUM WORD

013726  062003                  3$:     ADD     (R0)+,R3        ;FORM SUM
013730  005503                          ADC     R3              ;WITH CARRY
013732  005301                          DEC     R1              ;MORE WORDS?
013734  001374                          BNE     3$              ;YES

013736  032737  000001  003310          BIT     #BIT0,SYSTAT    ;WAS IT ODD
013744  001405                          BEQ     4$              ;NO
013746  112002                          MOVB    (R0)+,R2        ;YES GET NEXT BYTE
013750  042702  177400                  BIC     #177400,R2      ;UN-SIGN EXTEND
013754  060203                          ADD     R2,R3           ;ADD IT IN
013756  005503                          ADC     R3              ;AND CARRY JUST IN CASE

013760  010301                  4$:     MOV     R3,R1           ;RETURN IT IN CORRECT PLACE
013762                                  POP     R2              ;RESTORE
013762  012602                                   MOV     (SP)+,R2
013764                                  POP     R3
013764  012603                                   MOV     (SP)+,R3
013766  000207                          RETURN                  ;RETURN
```

```
                                .SBTTL  CKCKSM / MODULE TO CHECK THE CHKSUMS

                                ;++
                                ; MAKE SURE THE CHECKSUM RECEIVED = THE CHECKSUM CALCULATED.
                                ; INPUTS:  R4 = THE PACKET BYTE COUNT
                                ;          R0 -> THE PACKET TOP
                                ; OUTPUTS:  CARRY SET IF CHECKSUM CALC'D DOES NOT EQUAL CHECKSUM SENT
                                ;           R0 -> THE PACKET TOP
                                ;--

013770                          CKCKSM:: PUSH   R1
(1) 013770  010146                               MOV     R1,-(SP)
(1)
(1)
013772                                  PUSH    R0              ;SAVE
(1) 013772  010046                              MOV     R0,-(SP)
(1)
(1)
013774  010401                          MOV     R4,R1           ;COPY BYTE COUNT TO CORRECT
013776  004737  013674                  CALL    CHKSUM          ;REGISTER FOR CHKSUM AND
                                                                ;FORM CHECKSUM

                                ;HERE R0 --> XMITTED CHKSUM, R1=CHKSUM CALC'D

014002  122001                          CMPB    (R0)+,R1        ;LOWER ORDER CHECK
014004  001005                          BNE     2$              ;WRONG

014006  000301                          SWAB    R1              ;OK-PREP FOR

014010  122001                          CMPB    (R0)+,R1        ;HIGH ORDER CHECK
014012  001002                          BNE     2$              ;WRONG
014014  000241                          CLC                     ;OK-CLEAR SAILING

014016  000401                          BR      3$              ;EXIT

014020  000261                  2$:     SEC                     ;LET ERROR BE KNOWN

014022                          3$:     POP     R0
(1) 014022  012600                               MOV     (SP)+,R0
(1)
014024                                  POP     R1
(1) 014024  012601                               MOV     (SP)+,R1
(1)
014026  000207                          RETURN                  ;RETURN
```

```
 6194                                   .SBTTL  DOBRK / MODULE TO INIT TU58 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:  @R5 BIT14 WAS SET - (SEND BREAK)
 6207                                   ; OUTPUTS: @R5 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  014030  105037  014563           DOBRK:: CLRB    INITWD+1        ;CLEAR BYTE RECEIVE ADDR
 6217  014034  005037  014564                   CLR     BRKTO           ;CLEAR TIME OUT CONSTANT
 6218  014040  052775  000001  000026           BIS     #BIT0,@XMSR(R5) ;SET 'BREAK'
 6219  014046  012765  000001  000100           MOV     #RSSNIT,CMDSNT(R5)  ;SAY WE SENT 'INIT'
 6220  014054  052715  000020                   BIS     #BIT4,@R5       ;PAK SENT TYPE =COMMAND, SORT OF
 6221  014060  012704  000010                   MOV     #8.,R4          ;BREAK-IT'S-BACK COUNT=8
 6222  014064                           1$:     BREAK                   ;SUPERVISOR TAKE FIVE
  (3)  014064  104422                                                                              TRAP    C$BRK
 6223                                                                   ;FOR ^C CHECK, ETC.
 6224  014066  105775  000026                   TSTB    @XMSR(R5)       ;READY?
 6225  014072  100410                           BMI     4$              ;YES
 6226  014074  005337  014564                   DEC     BRKTO           ;NO,TIME OUT?
 6227  014100  001371                           BNE     1$              ;NO
 6228  014102  012704  000056                   MOV     #TOSNDB,R4      ;YES, SET ERROR CODE
 6229  014106  004737  012654                   CALL    LOG             ;LOG IT
 6230  014112  000535                           BR      3$              ;EXIT
 6231  014114  113775  014560  000030   4$:     MOVB    BRKWD,@XMDB(R5) ;SEND NULL
 6232  014122  005037  014564                   CLR     BRKTO           ;RESET TIME OUT
 6233  014126  005304                           DEC     R4              ;MORE NULLS TO SEND?
 6234  014130  001355                           BNE     1$              ;YES
 6235  014132  005375  000026                   DEC     @XMSR(R5)       ;NO, CLEAR 'BREAK'
 6236  014136  017500  000024                   MOV     @RCDB(R5),R0    ;HEAVE 'GARBAGE' 1ST BYTE
 6237  014142                                   SETPRI  #PRI00          ;SET TO INTERRUPT FO SURE
  (3)  014142  012700  000000                                                                      MOV     #PRI00,R
  (3)  014146  104441                                                                              TRAP    C$SPRI
 6238  014150                                   SETVEC  TUVECT(R5),#RCVINT,#PRI07 ;SET VECTO INFO
  (7)  014150  012746  000340                                                                      MOV     #PRI07,-
  (6)  014154  012746  014470                                                                      MOV     #RCVINT,
  (5)  014160  016546  000204                                                                      MOV     TUVECT(R
  (4)  014164  012746  000003                                                                      MOV     #3,-(SP)
  (3)  014170  104437                                                                              TRAP    C$SVEC
  (2)  014172  062706  000010                                                                      ADD     #10,SP
 6239  014176  062765  000004  000204           ADD     #4,TUVECT(R5)   ;AND INC TO SND VECTOR
 6240  014204                                   SETVEC  TUVECT(R5),#SNDINT,#PRI07;AND SET IT
```

```
  (7) 014204 012746 000340                                                                MOV     #PRIO7,
  (6) 014210 012746 014454                                                                MOV     #SNDINT,
  (5) 014214 016546 000204                                                                MOV     TUVECT(R
  (4) 014220 012746 000003                                                                MOV     #3,-(SP)
  (3) 014224 104437                                                                       TRAP    C$SVEC
  (2) 014226 062706 000010                                                                ADD     #10,SP
6241  014232 162765 000004 000204          SUB     #4,TUVECT(R5)   ;RESET VECTOR ADDR.
6242  014240 005037 014564                CLR     BRKTO           ;RESET TIME OUT
6243  014244 012704 014562                MOV     #INITWD,R4      ;USE ADDR. FOR SNDBYT
6244  014250 010437 014566                MOV     R4,BRKPTR       ;AND SAVE FOR "WAIT"
6245  014254 052775 000100 000026          BIS     #BIT6,@XMSR(R5) ;ENABLE INTER.
6246  014262 004737 014524                CALL    WAIT            ;AND ENTER LOOP
6247  014266 005715                       TST     @R5             ;ABORTED FROM TIME OUT?
6248  014270 100446                       BMI     3$              ;YES-EXIT
6249
6250  014272 005037 014564                CLR     BRKTO           ;RESET TIME OUT
6251  014276 012704 014562                MOV     #INITWD,R4      ;SEND SECOND INIT
6252  014302 010437 014566                MOV     R4,BRKPTR       ;SAVE POINTER AGAIN
6253  014306 052775 000100 000026          BIS     #BIT6,@XMSR(R5) ;AND THEN ENABLE INT
6254  014314 004737 014524                CALL    WAIT            ;AND WAIT
6255  014320 005715                       TST     @R5             ;IF ABORTED
6256  014322 100431                       BMI     3$              ;THEN EXIT
6257
6258  014324 012704 014563                MOV     #INITWD+1,R4    ;WHERE RESPONSE WILL GO (ADDRESS)
6259  014330 010437 014566                MOV     R4,BRKPTR       ;AND FOR 'WAIT'
6260  014334 052775 000100 000022          BIS     #BIT6,@RCSR(R5) ;ENABLE RECIEVE INT.
6261  014342 004737 014524                CALL    WAIT            ;GET ANSWER
6262  014346 005715                       TST     @R5             ;ABORTED?
6263  014350 100416                       BMI     3$              ;YES.
6264
6265  014352 123727 014563 000020          CMPB    INITWD+1,#RSCONT  ;NO, IS IT 'CONTINUE'?
6266  014360 001003                       BNE     2$              ;NOPE-ERROR
6267
6268  014362 042715 040000                BIC     #BIT14,@R5      ;SUCCESSFUL, CLEAR DOBREAK FLAG
6269  014366 000407                       BR      3$              ;EXIT
6270
6271  014370 113737 014563 003311  2$:    MOVB    INITWD+1,SYSTAT+1       ;SAVE BUM RESPONSE
6272  014376 012704 000032                MOV     #CNINIT,R4              ;CAN'T INIT CODE
6273  014402 004737 012654                CALL    LOG             ;LOG IT
6274                                                              ;SCHEDULER WILL TRY AGAIN IF NOT ABORTED
6275
6276  014406 042775 000100 000026  3$:    BIC     #BIT6,@XMSR(R5) ;CLEAR INTERRUPTS
6277  014414 042775 000100 000022          BIC     #BIT6,@RCSR(R5) ; AND FOR RECIEVE
6278  014422                              CLRVEC  TUVECT(R5)      ;RELEASE RECIEVE VECT.
  (3) 014422 016500 000204                                                                MOV     TUVECT(R
  (3) 014426 104436                                                                       TRAP    C$CVEC
6279  014430 062765 000004 000204          ADD     #4,TUVECT(R5)   ;AND GET SEND ADDR.
6280  014436                              CLRVEC  TUVECT(R5)      ;AND RELEASE IT
  (3) 014436 016500 000204                                                                MOV     TUVECT(R
  (3) 014442 104436                                                                       TRAP    C$CVEC
6281  014444 162765 000004 000204          SUB     #4,TUVECT(R5)   ;RESTORE POINTER
6282  014452 000207                       RETURN                  ;RETURN
```

```
                                        .SBTTL  INTERRUPT SERVICE ROUTINES AND TIMER
                                        BGNSRV  SNDINT                  ;"SEND" INTERRUPT SERVICE:
                                                                                        SNDINT::
                                        SNDMND: BIC     #BIT6,@XMSR(R5) ;DISABLE INTERRUPT
                                                MOVB    (R4)+,@XMDB(R5);OUTPUT BYTE
                                        ENDSRV
                                                                                        L10004:
                                                                                                RTI

                                        BGNSRV  RCVINT                  ;"RCV" INTERRUPT SERVICE:
                                                                                        RCVINT::
                                        RCVMND: BIC     #BIT6,@RCSR(R5) ;DISABLE INTS
                                                MOV     @RCDB(R5),DLV(R5)       ;SAVE WORD
                                                MOVB    DLV(R5),(R4)+   ;BYTE TO BUFFER
                                                TST     DLV(R5)         ;ERROR?
                                                BMI     10$             ;YES
                                                CLR     DLV(R5)         ;NO CLEAR ERROR
                                        10$:
                                        ENDSRV
                                                                                        L10005:
                                                                                                RTI

                                        WAIT:   NOP                     ;WAIT  LOOP FOR
                                                                        ;INTERRUPT SERVICING
                                                CMP     R4,BRKPTR       ;IF=,THEN NO INTERRUPT
                                                BNE     1$              ;GOT ONE!
                                                BREAK                   ;SUPERVISOR BREAK
                                                                                        TRAP    C$BRK
                                                BREAK                   ;KILL SOME TIME
                                                                                        TRAP    C$BRK
                                                DEC     BRKTO           ;TIME OUT?
                                                BNE     WAIT            ;NO...CONT.
                                                MOV     #TORCVB,R4      ;YES LOAD ERROR #
                                                CALL    LOG             ;LOG IT
                                        1$:     RETURN                  ;RETURN

                                        BRKWD:  .WORD   0               ;NULL
                                        INITWD: .BYTE   RSINIT          ;INIT COMMAND
                                                .BYTE   0               ;RSCONT IS EXPECTED HERE
                                        BRKTO:  .WORD   0               ;TIME OUT
                                        BRKPTR: .WORD   0               ;POINTER TO INITWD
```

```
                                .SBTTL  COMPAR/DATA COMPARISON MODULE

                                ;++
                                ; COMPAR - IF "COMPARE_DATA" SELECTED, COMPARE EACH DATA BYTE OF PACKET
                                ;          TO PATTEN(R5).  SAVE NUMBER OF BYTES NOT CORRECT.  IF NOT
                                ;          0, PRINT SOFT ERROR AND TOTAL # WRONG BYTES.  SET "BAD_DATA_
                                ;          IN_PACKET" BIT (BIT6 @R5) FOR HIGHER LEVEL MODULES.
                                ; INPUTS:  - (CMPDAT) FLAG TO NOT COMPARE (=1)
                                ;          - PKPTR(R5) POINTS TO DATA PACK.
                                ; OUTPUTS: BIT6 @R5 (BAD DATA FLAG) ADJUSTED.
                                ;          L$LUN - UNIT NUMBER
                                ;          PRNSIZ - SIZE OF PACKET
                                ;--

014570                          COMPAR:: PUSH   R0              ;COMPARE DATA IS DATA PACKET
014570  010046                                                  MOV     R0,-(SP)


014572                                   PUSH   R4              ;TO PATTERN WRITTEN
014572  010446                                                  MOV     R4,-(SP)


014574                                   PUSH   R1              ;USING BYTE COUNT IN PACKET
014574  010146                                                  MOV     R1,-(SP)


014576  005037  014746                     CLR    BDBYTS          ;CLEAR TOTAL WRONG
014602  016304  000104                     MOV    PKPTR(R5),R4    ;GET TOP OF PACKET
014606  005737  002216                     TST    CMPDAT          ;COMPARE SELECTED?
014612  001451                           BEQ    4$              ;NO-EXIT
014614  005204                           INC    R4              ;YES, LOCATE COUNT
014616  111401                           MOVB   @R4,R1          ;GET IT
014620  042701  177400                   BIC    #177400,R1      ;SIGN-UNEXTEND
                                                                ;MUST TEST BYTE-WISE...
014624  005204                           INC    R4              ;-->FIRST DATA  BYTE
014626  126524  000072                1$:    CMPB   PATTEN(R5),(R4)+  ;DATA-WHAT WAS EXPECTED?
014632  001402                           BEQ    2$              ;YES
014634  005237  014746                     INC    BDBYTS          ;NO, INCREMENT TOTAL WRONG
014640  005301                        2$:    DEC    R1              ;MORE LEFT?
014642  001371                           BNE    1$              ;YES
014644  005737  014746                     TST    BDBYTS          ;ANY WRONG?
014650  001432                           BEQ    4$              ;NO
014652  011537  002074                     MOV    @R5,L$LUN       ;GET UNIT NUMBER
014656  042737  177770  002074           BIC    #177770,L$LUN   ;MASK IT OFF
014664                                   ERRSOFT 0.,MSBDA,ERRDES  ;YES-PRINT 'BAD DATA IN PACKET' ERROR
014664  104457                                                          TRAP    C$ERSOFT
014666  000000                                                          .WORD   0
014670  002342                                                          .WORD   MSBDA
014672  013210                                                          .WORD   ERRDES
014674                                   PRINTB  #DESC,BDBYTS
014674  013746  014746                                                  MOV     BDBYTS,-
014700  012746  014750                                                  MOV     #DESC, (
014704  012746  000002                                                  MOV     #2,-(SP)
014710  010600                                                          MOV     SP,R0
014712  104414                                                          TRAP    C$PNTB
014714  062706  000006                                                  ADD     #6,SP
```

```
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 008

 6364  014720  052715  000100                       BIS     #BIT6,@R5       ;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                           4$:         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
```

```
                                        .SBTTL  PRNPAK/MODULE TO PRINT DATA PACKET

                                        ;++
                                        ; PRNPAK - IF PRINT_DATA_PACK_ON_ERROR SELECTED: PRINT EACH BYTE OF PACKET
                                        ;          TO BY PKPTR(R5).
                                        ; INPUTS:  PRNSIZ - # OF BYTES IN PACKET.
                                        ; OUTPUTS: NONE
                                        ;--

015004  000240                          PRNPAK:: NOP                    ;PRINTS 1 PACKET
                                                                        ;PKPTR(R5)->TOP OF PACKET
                                                                        ;PRNSIZ (PASSED)=BYTE COUNT
015006                                          PUSH    R0
015006  010046                                                  MOV     R0,-(SP)
015010                                          PUSH    R4
015010  010446                                                  MOV     R4,-(SP)
015012  105737  002214                          TSTB    PRBUF           ;PRINT PACKET SELECTED?
015016  001451                                  BEQ     4$              ;NO
015020  016504  000104                          MOV     PKPTR(R5),R4    ;YES-GET TOP OF PACK
015024  012737  000020  015150          1$:     MOV     #16.,LNCNT      ;16 BYTES PER LINE
015032  112437  015152                  2$:     MOVB    (R4)+,PRDAT     ;AVOID SIGN EXTEND
015036                                          PRINTF  #PRFORM.<8,PRDAT>  ;PRINT BYTE
015036  005046                                                                          CLR     -(SP)
015040  153716  015152                                                                  BISB    PRDAT,(S
015044  012746  015154                                                                  MOV     #PRFORM,
015050  012746  000002                                                                  MOV     #2.-(SP)
015054  010600                                                                          MOV     SP,R0
015056  104417                                                                          TRAP    C$PNTF
015060  062706  000006                                                                  ADD     #6.SP
015064  005337  003340                          DEC     PRNSIZ          ;ONE LESS
015070  001414                                  BEQ     3$              ;NO MORE
015072  005337  015150                          DEC     LNCNT           ;NEW LINE?
015076  001355                                  BNE     2$              ;NOT YET
015100                                          PRINTF  #CARLF          ;YES
015100  012746  015164                                                                  MOV     #CARLF,-
015104  012746  000001                                                                  MOV     #1.-(SP)
015110  010600                                                                          MOV     SP,R0
015112  104417                                                                          TRAP    C$PNTF
015114  062706  000004                                                                  ADD     #4.SP
015120  000741                                  BR      1$              ;NEXT LINE
015122                                  3$:     PRINTF  #CARLF          ;FINISH UP
015122  012746  015164                                                                  MOV     #CARLF,-
015126  012746  000001                                                                  MOV     #1.-(SP)
015132  010600                                                                          MOV     SP,R0
015134  104417                                                                          TRAP    C$PNTF
015136  062706  000004                                                                  ADD     #4.SP
015142                                  4$:     POP     R4
015142  012604                                                  MOV     (SP)+,R4
015144                                          POP     R0
015144  012600                                                  MOV     (SP)+,R0
```

```
 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                                        ENDMOD
 6417
```

```
                                        .TITLE MISCELLANEOUS SECTIONS
                                        .SBTTL  REPORT CODING SECTION

                                                BGNMOD

                                        ;++
                                        ; THE REPORT CODING SECTION CONTAINS THE
                                        ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                        ;--

                                                BGNRPT                                          L$RPT::
                                                PUSH    R0
                                                                        MOV     R0,-(SP)
                                                PUSH    R1
                                                                        MOV     R1,-(SP)
                                                PUSH    R2
                                                                        MOV     R2,-(SP)
                                                PUSH    R3
                                                                        MOV     R3,-(SP)
                                                PUSH    R4
                                                                        MOV     R4,-(SP)
                                                PUSH    R5
                                                                        MOV     R5,-(SP)

                                                BREAK                                                   TRAP    C$BRK
                                                MOV     @BLKTBL,RPTR    ;GET 1ST DEVICE BLOCK
                                                PRINTS  @STATHD         ;HEADER
                                                                                                        MOV     @STATHD,
                                                                                                        MOV     #1,-(SP)
                                                                                                        MOV     SP,R0
                                                                                                        TRAP    C$PNTS
                                                                                                        ADD     #4,SP
                                                BREAK                   ;^C CHECK
                                                                                                        TRAP    C$BRK
                                                PRINTS  @STHD2          ;2ND HEADER
                                                                                                        MOV     @STHD2,-
                                                                                                        MOV     #1,-(SP)
                                                                                                        MOV     SP,R0
                                                                                                        TRAP    C$PNTS
                                                                                                        ADD     #4,SP
                                        1$:     BREAK                   ;^C CHECK                       TRAP    C$BRK
```

```
 6481  015260  017705  000332                    MOV     @RPTR,R5        ;GET DEVICE BLOCK
 6482  015264  032715  004000                    BIT     #BIT11,@R5      ;UNIT NOT TESTED?
 6483  015270  001131                            BNE     2$              ;TRUE, DON'T PRINT STATISTICS
 6484                                                                    ;OK TO PRINT
 6485  015272  011537  015614                    MOV     @R5,RLUN        ;SAVE STATUS WORD
 6486  015276  042737  177770  015614            BIC     #177770,RLUN    ;MASK UNIT NUM.
 6487  015304  116501  000122                    MOVB    SOFTR(R5),R1    ;SOFTREAD
 6488  015310  042701  177400                    BIC     #177400,R1      ;SIGN-UNEXTEND
 6489  015314  116502  000124                    MOVB    SOFTW(R5),R2    ;SOFT WRITE
 6490  015320  042702  177400                    BIC     #177400,R2      ;
 6491  015324  116503  000136                    MOVB    HARDR(R5),R3    ;HARD READ
 6492  015330  042703  177400                    BIC     #177400,R3      ;
 6493  015334  116504  000140                    MOVB    HARDW(R5),R4    ;HARD WRITE
 6494  015340  042704  177400                    BIC     #177400,R4      ;
 6495  015344                                    PRINTS  #FMO,RLUN       ;SUMMARY/UNIT #
  (8)  015344  013746  015614                                                            MOV     RLUN,-(S
  (7)  015350  012746  015732                                                            MOV     #FMO,-(S
  (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                    MOVB    SOFTR+1(R5),R1  ;SAME
 6498  015446  042701  177400                    BIC     #177400,R1      ;AS
 6499  015452  116502  000125                    MOVB    SOFTW+1(R5),R2  ;ABOVE
 6500  015456  042702  177400                    BIC     #177400,R2      ;THIS
 6501  015462  116503  000137                    MOVB    HARDR+1(R5),R3  ;TIME
 6502  015466  042703  177400                    BIC     #177400,R3      ;FOR
 6503  015472  116504  000141                    MOVB    HARDW+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                                                                                    MOV     #11,-(SP
   (3)  015544  010600                                                                                            MOV     SP,R0
   (4)  015546  104416                                                                                            TRAP    C$PNTS
   (4)  015550  062706  000024                                                                                    ADD     #24,SP
 6507   015554  023727  015616  003370          2$:     CMP     RPTR,#LSTDEV    ;ALL UNITS DONE?
 6508   015562  103005                                  BHIS    3$              ;YES
 6509   015564  062737  000002  015616                  ADD     #2,RPTR         ;NO-DO
 6510
 6511   015572  000137  015256                          JMP     1$              ;MORE UNITS
 6512
 6513   015576                                  3$:     POP     R5
   (1)  015576  012605                                                          MOV     (SP)+,R5
   (1)                                  ,
 6514   015600                                          POP     R4
   (1)  015600  012604                                                          MOV     (SP)+,R4
   (1)
 6515   015602                                          POP     R3
   (1)  015602  012603                                                          MOV     (SP)+,R3
   (1)
 6516   015604                                          POP     R2
   (1)  015604  012602                                                          MOV     (SP)+,R2
   (1)
 6517   015606                                          POP     R1
   (1)  015606  012601                                                          MOV     (SP)+,R1
   (1)
 6518   015610                                          POP     R0
   (1)  015610  012600                                                          MOV     (SP)+,R0
   (1)
 6519   015612                                          ENDRPT
   (3)  015612                                                                                    L10006:
   (3)  015612  104425                                                                                    TRAP    C$RPT
 6520   015614  000000                          RLUN:   .WORD
 6521   015616  000000                          RPTR:   .WORD
 6522
 6523   015620  047045  040445  020040          STATHD: .ASCII  /#N#A      DR BLKS WR  BLKS RD  BDPAK  /
 6524   015666  041504  045510  051057                  .ASCIZ  @DCHK/RD DCHK/WR  DCHK/RD DCHK/WR#N@
 6525           015732                                  .EVEN
 6526   015732  040445  047125  052111          FM0:    .ASCIZ  /#AUNIT #D1#N/
 6527           015750                                  .EVEN
 6528
 6529   015750  040445  020040  020040          FM:     .ASCII  /#A       #D1#A #D5#A.   #D5#A.    #D3#A.    /
 6530   016024  042045  022463  027101                  .ASCIZ  /#D3#A.   #D3#A.     #D3#A.    #D3#A.#N/
 6531           016074                                  .EVEN
 6532   016074  040445  020040  020040          STHD2:  .ASCII  /#A                                    /
 6533   016141     122  041505  053117                  .ASCIZ  /RECOV   RECOV   UNRECOV UNRECOV#N/
 6534           016204                                  .EVEN
 6535   016204                                          ENDMOD
```

```
                                        .SBTTL  INITIALIZE SECTION

                                        ;++
                                        ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                        ; AT THE BEGINNING OF EACH PASS.
                                        ;

016204                                          BGNINIT
016204                                                                                          L$INIT::
016204  000240                  INIT:   NOP                     ;
016206  105037  016770                  CLRB    STRT            ;FOR STATS CLEAR
016212  005037  003344                  CLR     TEST9           ;***** CLR TST 9 FLAG
016216                                  READEF  #EF.START       ;START COMMAND?
016216  012700  000040                                                          MOV     #EF.STAR
016222  104447                                                                  TRAP    C$REFG
016224                                  BNCOMPLETE INIT2        ;NO
016224  103003                                                                  BCC     INIT2
016226  005237  016770                  INC     STRT            ;YES, SET START FLAG
016232                                  BRESET                  ;BUSS RESET, EH?
016232  104433                                                                  TRAP    C$RESET
016234  012737  003352  003314  INIT2:  MOV     #BLKTBL,DEVPTR  ;SET ALL UNITS ABORTED:
016242  005004                          CLR     R4              ;UNIT NUMBER
016244  017705  165044          1$:     MOV     @DEVPTR,R5      ;GET POINTER
016250  010415                          MOV     R4,@R5          ;INSERT UNIT #
016252  052715  100000                  BIS     #BIT15,@R5        ;SET ABORTED
016256  052715  004000                  BIS     #BIT11,@R5      ;SET UNIT NOT TESTED
016262  006304                          ASL     R4              ;*2 FOR LOOK-UP
016264  016465  027724  000102          MOV     BUFTBL(R4),RCVBUF(R5)  ;SETUP POINTER TO UNIT'S BUFFER
016272  006204                          ASR     R4              ;CORRECT BACK TO UNIT #
016274  023727  003314  003370          CMP     DEVPTR,#LSTDEV  ;LAST DEVICE DONE?
016302  103005                          BHIS    CHECK           ;YES
016304  062737  000002  003314          ADD     #2,DEVPTR       ;NO-GET
016312  005204                          INC     R4              ;NEXT DEVICE AND
016314  000753                          BR      1$              ;SERVICE

016316  022737  000010  002012  CHECK:  CMP     #8.,L$UNIT      ;MAKE SURE NOT
016324  103005                          BHIS    GETHRD          ;TOO MANY UNITS
016326                                  ERRSF   101.,TOMANY             ;TOMANY-REQUEST TO
016326  104454                                                          TRAP    C$ERSF
016330  000145                                                          .WORD   101
016332  016706                                                          .WORD   TOMANY
016334  000000                                                          .WORD   0
016336                                  DOCLN                   ;EXIT
016336  104444                                                          TRAP    C$DCLN

016340  012737  003352  003314  GETHRD: MOV     #BLKTBL,DEVPTR  ;INIT TABLE POINTER
016346  005004                          CLR     R4              ;CLEAR DEVICE COUNTER
016350  017705  164740          1$:     MOV     @DEVPTR,R5      ;GET STATUS WORD
016354  010437  002074                  MOV     R4,L$LUN        ;UNIT NUM. IN CASE ERROR
016360                                  GPHARD  R4,R2           ;GET HARD INFO
016360  010400                                                          MOV     R4,R0
016362  104442                                                          TRAP    C$GPHRD
016364  010002                                                          MOV     R0,R2
016366                                  BNCOMPLETE 3$
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$ERSF
  (5)  016442  000146                                                                .WORD   102
  (5)  016444  016736                                                                .WORD   NODRVS
  (5)  016446  000000                                                                .WORD   0
 6595  016450                                        DOCLN                   ;EXIT
  (3)  016450  104444                                                                TRAP    C$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  000030              4$:      MOV     R3,XMDB(R5)
 6619  016556  005065  000072                        CLR     PATTEN(R5)      ;ZERO DATA PATTERN
 6620  016562  005065  000002                        CLR     RETRY(R5)       ;NO RETRIES
 6621  016566  005065  000064                        CLR     REC(R5)         ;NO RECORD
 6622  016572  005065  000076                        CLR     SUCCS(R5)       ;NO SUCCESS
 6623  016576  005065  000074                        CLR     DLV(R5)         ;NO DLV ERROR
 6624  016602  005065  000210                        CLR     MRSP(R5)        ;***** CLR MRSP INDICATOR
 6625  016606  005037  003342                        CLR     ALLGON          ;OK TO PRINT STATISTICS
 6626  016612  062737  000002  003314        3$:     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  FLGLOC          ;GET USER FLAGS
  (3)  016634  104421                                                                TRAP    C$RFLA
```

```
  (3)  016636  010037  016774                                                                    MOV     R0,FLGLO
6633   016642  005037  003334          5$:     CLR     BLKER           ;NO ERROR
6634   016646  013737  002210  003312  SETLEN: MOV     LENGTH,TAPLEN   ;GET # OF RECORDS
6635   016654  006237  003312          ASR     TAPLEN                  ;GET # BLOCKS PER TRACK
6636   016660  012737  000200  003336  MOV     #200,SECREC             ;PRESET SECOND START AT 200
6637   016666  022737  000200  003312  CMP     #200,TAPLEN             ;# BLKS > 128.?
6638   016674  101003                  BHI     3$                      ;NO-SWITCH TRACKS 2ND PASS
6639   016676  012737  000400  003336  MOV     #400,SECREC             ;YES-START AT 400
6649
6661
6662   016704                          3$:     ENDINIT
  (3)  016704                                                                          L10007:
  (3)  016704  104411                                                                            TRAP    C$INIT
6663
6664
6665   016706  047524  020117  040515  TOMANY: .ASCIZ  /TOO MANY UNITS MAX.=8 /
6666           016736                  .EVEN
6667   016736  042523  042514  052103  NODRVS: .ASCIZ  /SELECT AT LEAST 1 DRIVE /
6668           016770                  .EVEN
6669   016770  000000                  STRT::  .WORD
6670   016772  000000                  PDTFLG::.WORD           ;TU58 IS IN PDT
6671   016774  000000                  FLGLOC::.WORD           ;USER FLAGS
```

```
 6674                                                ;**
 6675                                                ; THE AUTO DROP CODE IS INVOKED WHEN THE ADR FLAG IS SET AND CHECKS FOR
 6676                                                ; A VALID INTERFACE LOCATION.  DROPS UNIT IF INTERFACE IS NOT THERE.
 6677                                                ; -
 6678
 6679  016776                                        BGNAUTO
  (3)  016776                                                                                    L$AUTO::
 6680  016776  000240                                       NOP             ;AUTO DROP ROUTINE
 6681  017000                                               SETVEC  #4,#TRPHND,#PRIO7       ;GET BUS TRAP VEC.
  (7)  017000  012746  000340                                                                                MOV     #PRIO7,
  (6)  017004  012746  017106                                                                                MOV     #TRPHND,
  (5)  017010  012746  000004                                                                                MOV     #4,(SP)
  (4)  017014  012746  000003                                                                                MOV     #3,-(SP)
  (3)  017020  104437                                                                                        TRAP    C$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$:     MOV     @TRPPTR,R5      ;GET DATA BLOCK
 6684  017040  032715  104000                         BIT     #BIT15!BIT11,@R5  ;NOT TESTED OR ABORTED?
 6685  017044  100403                                 BMI     2$              ;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$:     CMP     TRPPTR,#LSTDEV  ;MORE TO TRY?
 6689  017062  103004                                 BHIS    3$              ;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                                                                                    L10010:
  (3)  017102  104461                                                                                        TRAP    C$AUTO
 6694  017104  000000                        TRPPTR: .WORD
 6695
 6696
 6697
 6698
 6699                                                ;ILLEGAL ADDRESS TRAP HANDLER:
 6700
 6701  017106                                 TRPHND: PRINTF  #MSAUTO         ;SAY "AUTO DROPPED"
  (7)  017106  012746  017140                                                                                MOV     #MSAUTO,
  (6)  017112  012746  000001                                                                                MOV     #1,-(SP)
  (3)  017116  010600                                                                                        MOV     SP,R0
  (4)  017120  104417                                                                                        TRAP    C$PNTF
  (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                                 RTI
 6706  017140  040445  052501  047524         MSAUTO: .ASCIZ  /%AAUTO DROP: %N/
```

```
 6709                                           .SBTTL  CLEANUP CODING SECTION
 6710
 6711                                           ;++
 6712                                           ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
 6713                                           ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
 6714                                           ;--
 6715
 6716  017160                                           BGNCLN
  (3)  017160                                                                                  L$CLEAN::
 6717  017160  005737  003342                          TST     ALLGON          ;ENTRANCE FROM ALL-UNITS-ABORTED?
 6718  017164  001004                                  BNE     1$              ;YES-EXIT
 6719  017166  005737  002212                          TST     STAEOP          ;NO-STATS AT EOP?
 6720  017172  001401                                  BEQ     1$              ;NO
 6721  017174                                          DORPT                   ;YES
  (3)  017174  104424                                                                          TRAP    C$DRPT
 6722
 6729
 6741
 6742  017176                                  1$:     BRESET
  (3)  017176  104433                                                                          TRAP    C$RESET
 6743  017200                                          ENDCLN
  (3)  017200                                                                          L10011:
  (3)  017200  104412                                                                          TRAP    C$CLEAN
```

```
                                        .SBTTL  DROP UNIT SECTION

                                        ;**
                                        ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                        ; TO NO LONGER BE TESTED.
                                        ;-

                                                BGNDU
                                                                                        L$DU::
                                                                        ;R0=UNIT NUMBER
017202                                          PUSH    R0              ;SAVE IT
(1) 017202  010046                                                      MOV     R0,-(SP)
(1)
(1)
017204                                          PUSH    R5              ;SAVE PRESENT UNIT POINTER
(1) 017204  010546                                                      MOV     R5,-(SP)
(1)
(1)
017206  004737  017246                          CALL    GETR5           ;GET POINTER TO UNIT
017212  052715  100000                          BIS     #BIT15,@R5      ;SET ABORTED
017216                                          POP     R5              ;RESTORE PRESENT UNIT POINTER
(1) 017216  012605                                                      MOV     (SP)+,R5
(1)
017220                                          POP     R0              ;RETRIEVE UNIT NUMBER
(1) 017220  012600                                                      MOV     (SP)+,R0
(1)
017222                                          PRINTF  #ABOMSG,R0
(8) 017222  010046                                                                              MOV     R0,-(SP)
(7) 017224  012746  017300                                                                      MOV     #ABOMSG,
(6) 017230  012746  000002                                                                      MOV     #2, (SP)
(3) 017234  010600                                                                              MOV     SP,R0
(4) 017236  104417                                                                              TRAP    C$PNTF
(4) 017240  062706  000006                                                                      ADD     #6,SP

                                                ENDDU
017244                                                                                  L10012:
(3) 017244                                                                                      TRAP    C$DU
(3) 017244  104453

017246  012737  003352  017276    GETR5:  MOV     #BLKTBL,PTR     ;-->UNIT 0
017254  017705  000016            1$:     MOV     @PTR,R5         ;GET STATUS WORD
017260  005300                            DEC     R0              ;CORRECT UNIT?
017262  100404                            BMI     2$              ;YES
017264  062737  000002  017276            ADD     #2,PTR          ;NO,-->NEXT
017272  000770                            BR      1$              ;CONTINUE
017274  000207                    2$:     RETURN
017276  000000                    PTR:    .WORD

                                        ABOMSG: .ASCIZ  /%ADROPPED UNIT %D1%N/
017300  040445  051104  050117            .EVEN
017326
```

```
 6795                                           .SBTTL  ADD UNIT SECTION
 6796
 6797                                           ;++
 6798                                           ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
 6799                                           ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
 6800                                           ;  TO THE TEST CYCLE.
 6801                                           ;-
 6802
 6803  017326                                           BGNAU
  (3)  017326                                                                    L$AU::
 6804
 6805                                           ;THE INIT CODE CONTAINS ALL CODE NECESSARY TO ADD A UNIT.
 6811
 6823
 6824
 6825
 6826  017326                                           ENDAU
  (3)  017326                                                                    L10013:
  (3)  017326  104452                                                                 TRAP    C$AU
 6827
```

```
 6886                                           .SBTTL  TEST 1 / DEVICE SELF-DIAGNOSTIC EXECUTION
 6887
 6888  017330                                   BGNMOD
 6889                                           .NLIST  ME,BEX
 6890
 6891  017330                                   BGNTST
  (3)  017330                                                                                    T1::
 6892  017330                                   TSTID   #TST1
  (1)  017330  012737  017374  003330                                    MOV     #TST1,TSTTOP    ;SAVE ADDR OF TEST
  (1)  017336  004737  006024                                            CALL    SETUP           ;INIT UNITS TSTPC
  (1)  017342  004737  005652                                            CALL    SETDR           ;GET 1ST DRVS.
  (1)  017346  004737  006072                                            CALL    RUN             ;DO TEST
  (1)  017352  004737  005532                                            CALL    SWAPDR          ;GET NEXT DRVS.
  (1)  017356  103004                                                    BCC     64$             ;BR NO 2ND DRVS
  (1)  017360  004737  006024                                            CALL    SETUP           ;REINIT UNITS TSTPC
  (1)  017364  004737  006072                                            CALL    RUN             ;REPEAT TEST
  (1)  017370                                                    64$:                            ;DONE
 6893  017370                                   EXIT TST
  (3)  017370  104432                                                                                    TRAP    C$EXIT
  (3)  017372  000136                                                                                    .WORD   L10014-.
 6894
 6895  017374                                   TST1:   TUSELF
  (1)
  (1)  017374  012700  027746                                    64$:    MOV     #TRBUF,R0       ;FORM COMMAND PACKET
  (1)  017400  112710  000002                                            MOVB    #RSCMND,@R0     ;COMMAND FLAG
  (1)  017404  112760  000012  000001                                    MOVB    #RSMSIZ,1(R0)   ;SIZE OF MESSAGE
  (1)  017412  112760  000007  000002                                    MOVB    #RSSSLF,2(R0)   ;SELF TEST OPERATION
  (1)  017420  105060  000003                                            CLRB    3(R0)           ;NO MODIFIER.
  (1)  017424  005060  000004                                            CLR     4(R0)           ;NO DRIVE OR SWITCHES
  (1)  017430  005060  000006                                            CLR     6(R0)           ;NO SEQUENCE NUMBER
  (1)  017434  005060  000010                                            CLR     8.(R0)          ;NO BYTES
  (1)  017440  005060  000012                                            CLR     10.(R0)         ;NO RECORD #
  (1)  017444  012701  000012                                            MOV     #RSMSIZ,R1      ;GET SIZE
  (1)  017450  005721                                                    TST     (R1)+           ;+2 FOR CHECKSUM
  (1)  017452  012765  000016  000070                                    MOV     #RSSNSZ,SNDCNT(R5)  ;SIZE TO SEND
  (1)  017460  004737  013674                                            CALL    CHKSUM          ;FORM CHECKSUM
  (1)  017464  010110                                                    MOV     R1,(R0)         ;INSERT INTO PACKET
  (1)  017466  012765  000002  000034                                    MOV     #RSEND,XSFLG(R5)  ;EXPECT END.
  (1)  017474  012765  000016  000076                                    MOV     #RSNDSZ,XSCNT(R5)  ;THIS BIG
  (1)  017502  012765  000001  000032                                    MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
  (1)                                                                                            ;SEND
  (1)  017510  004737  006600                                            CALL    RSVP            ;RETURN TO SCHEDULER
  (1)  017514  032715  000010                                            BIT     #BIT3,@R5       ;RETRY?(BAD FLAG)
  (1)  017520  001325                                                    BNE     64$             ;YES
  (1)
 6896  017522  005237  003324                   INC     DONE
 6897  017526  000207                           RETURN
 6898
 6899
 6900  017530                                   ENDTST
  (3)  017530                                                                                    L10014:
  (3)  017530  104401                                                                                    TRAP    C$ETST
```

```
                                        .SBTTL  TEST 2 / SEEK EOT,BOT

                                        BGNTST
                                                                                T2::
                                        TSTID   #TST2
                                                                MOV     #TST2,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
                                                        64$:                            ;DONE
                                        EXIT TST
                                                                                        TRAP    C$EXIT
                                                                                        .WORD   L10015-.

                                        TST2:   CLR     R4              ;R4=INDEX INTO RECORD TABLE
                                        1$:     MOV     RECDAT(R4),REC(R5)  ;GET THE RECORD

                                        TUSEEK  REC(R5),DR(R5)  ;SEEK IT

                                                        64$:    MOV     #TRBUF,R0       ;-->(POINT TO) XMIT BUFF
                                                                MOVB    #RSCMND,@R0     ;FORM COMMAND MESSAGE PA
                                                                MOVB    #RSMSIZ,1(R0)   ;THIS BIG
                                                                MOVB    #RSSSEK,2(R0)   ;OP CODE IS SEEK
                                                                MOV     REC(R5),10 (R0) ;TO THIS RECORD
                                                                MOVB    DR(R5),4.(R0)   ;AND WHICH DRIVE
                                                                CLRB    3.(R0)          ;NO MODIFIER
                                                                CLRB    5.(R0)          ;NO SWITCHES
                                                                CLR     6.(R0)          ;NO SEQUENCE #
                                                                CLR     8.(R0)          ;NO BYTE COUNT
                                                                MOV     #RSMSIZ,R1      ;GET COUNT
                                                                TST     (R1)+           ;PLUS FLAG + BCNT
                                                                                        ;FOR CHECKSUM CALC
                                                                CALL    CHKSUM          ;R0-->TOP   R1=0 OF BYTE
                                                                MOV     R1,(R0)         ;INSERT INTO PACKET
                                                                                        ;SET UP EXPECTATIONS:
                                                                MOV     #RSSNSZ,SNDCNT(R5)  ;HOW MANY TO SEND
                                                                MOVB    #RSCMND,XSFLG(R5)   ;EXPECT END PACK
                                                                MOV     #RSNDSZ,XSCNT(R5)   ;COUNT WITH THIS
                                                                MOV     #1.,XSPKNM(R5)  ;EXPECT ONLY 1 PACKET

                                                                CALL    RSVP            ;SEND
                                                                                        ;AND RETURN TO SCHEDULER
                                                                BIT     #BIT3,@R5       ;RETRY (FLAG BYTE ERROR)
                                                                BNE     64$             ;YES

                                        ADD     #2,R4           ;POINT TO NEXT RECORD
                                        CMP     RECDAT(R4),#-1. ;LAST ONE DONE?
                                        BNE     1$              ;NO-LOOP
                                        INC     DONE            ;YES-SET DONE FLAG
                                        RETURN
```

```
 6920
 6921  017766  000000                    RECDAT: 0.      ;BOT
 6922  017770  000200                            200     ;BOT OTHER TRACK
 6923  017772  000177                            177     ;EOT
 6924  017774  000377                            377     ;EOT OTHER TRACK
 6925  017776  000400                            400     ;BOT AGAIN
 6926  020000  177777                            -1.
 6927  020002                                    ENDTST
  (3)  020002                                                        L10015:
  (3)  020002  104401                                                        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.

                                        BGNTST
                                                                                        T3::
                                        TSTID   #TST3
020004  012737  020050  003330                          MOV     #TST3,TSTTOP    ;SAVE ADDR OF TEST
020012  004737  006024                                  CALL    SETUP           ;INIT UNITS TSTPC
020016  004737  005652                                  CALL    SETDR           ;GET 1ST DRVS.
020022  004737  006072                                  CALL    RUN             ;DO TEST
020026  004737  005532                                  CALL    SWAPDR          ;GET NEXT DRVS.
020032  103004                                          BCC     64$             ;BR NO 2ND DRVS
020034  004737  006024                                  CALL    SETUP           ;REINIT UNITS TSTPC
020040  004737  006072                                  CALL    RUN             ;REPEAT TEST
020044                                          64$:                            ;DONE
                                        EXIT TST
020044  104432                                                                    TRAP    C$EXIT
020046  001326                                                                    .WORD   L10016-.

020050  012765  000100  000066  TST3:   MOV     #100,TMP(R5)    ;INIT TO HALF OF REMAINING
020056  005004                          CLR     R4              ;FOR INDEX INTO DATA TABLE
020060  005065  000064                  CLR     REC(R5)         ;START AT RECORD 0
020064  016465  022766  000072  1$:     MOV     TST3PT(R4),PATTEN(R5)   ;GET DATA
020072                                  TUWRIT  PATTEN(R5),REC(R5),#512.,DR(R5),#0
020072  012700  027746                          MOV     #TRBUF,R0       ;MAKE COMMAND PACKET:
020076  112710  000002                          MOVB    #RSCMND,@R0     ;COMMAND FLAG
020102  112760  000012  000001                  MOVB    #RSMSIZ,1(R0)   ;THIS SIZE
020110  112760  000003  000002                  MOVB    #RSSWR,2(R0)    ;INSERT OP CODE-WRITE
020116  112760  000000  000003                  MOVB    #0,3.(R0)       ;VERIFY (1 OR 0)
020124  116560  000060  000004                  MOVB    DR(R5),4.(R0)   ;DRIVE #
020132  112760  000020  000005                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE SWITCH
020140  005060  000006                          CLR     6.(R0)          ;NO SEQUENCE #
020144  012760  001000  000010                  MOV     #512.,8.(R0)    ;TOTAL COUNT TO WRITE
020152  016560  000064  000012                  MOV     REC(R5),10.(R0) ;AT  RECORD N
020160  012701  000012                          MOV     #RSMSIZ,R1      ;THE PACKET SIZE PLUS*2
020164  005721                                  TST     (R1)+           ;(FLAG AND COUNT) INTO R
020166  012765  000016  000070                  MOV     #RSSNSZ,SNDCNT(R5)  ;LOAD THE  SIZE TO S
020174  004737  013674                          CALL    CHKSUM          ;R0 --> R1=COUNT
020200  010110                                  MOV     R1,(R0)         ;PUT CHKSUM IN PACKET
                                                                        ;SET UP EXPECTATIONS:
020202  012765  000020  000034                  MOV     #RSCONT,XSFLG(R5)   ;THE FLAG
020210  012765  000001  000036                  MOV     #1,XSCNT(R5)    ;THE COUNT
020216  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;THE # PACKETS EXPECTED
020224  012702  001000                          MOV     #512.,R2                ;GET # OF DATA B
020230  004737  006600                          CALL    RSVP            ;SEND (AND RETURN TO SCH
020234  032715  000010                          BIT     #BIT3,@R5       ;FLAG BYTE ERROR?
020240  001314                                  BNE     72$             ;YES
020242  042715  010000                          BIC     #BIT12,@R5      ;FLAG FOR LAST PACKET
020246  012700  027746                  64$:    MOV     #TRBUF,R0       ;POINT TO TOP OF BUFFER
020252  020227  000200                          CMP     R2,#128.        ;START DATA PACKET(S)
020256  101004                                  BHI     65$             ;#512. > 128.!
020260  010201                                  MOV     R2,R1           ;#512.<128.
020262  052715  010000                          BIS     #BIT12,@R5      ;SO LAST PACKET NOW
```

```
MISCELLANEOUS SECTIONS  MACY11 30(1046)  25-JAN-84  08:33  PAGE 53 1
CZTUUF.P11    25-JAN-84 08:09          TEST 3 / HIGH ACTIVITY WRITE/READ (512 BYTE/BLOCK MODE)                SEQ 0098

   (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$:    MOVB    PATTEN(R5),(R0)+        ;INSERT DATA
   (1)  020314  005303                                                        DEC     R3              ;MORE?
   (1)  020316  101374                                                        BHI     67$             ;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     #RSNDSZ,XSCNT(R5)  ;OF THIS SIZE
   (1)  020406  012765  000001  000032                                        MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
   (1)  020414  000411                                                        BR      69$             ;SEND
   (1)  020416  012765  000020  000034                                68$:    MOV     #RSCONT,XSFLG(R5)  ;(NOT LAST), EXPECT '
   (1)  020424  012765  000001  000036                                        MOV     #1,XSCNT(R5)    ;AND 1 BYTE
   (1)  020432  012765  000001  000032                                        MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
   (1)  020440  004737  006600                                        69$:    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 CMND PACK:
   (2)  020474  112710  000002                                                MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
   (2)  020500  112760  000012  000001                                        MOVB    #RSMSIZ,1(R0)   ;THIS BIG
   (2)  020506  112760  000002  000002                                        MOVB    #RSSRD,2(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  105060  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  012760  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     #RSMSIZ,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
```

```
  (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,XSPKNM(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)  020676  012700  027746                                    MOV     #TRBUF,R0       ;FORM CMND PACK:
  (1)  020702  112710  000002                                    MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
  (1)  020706  112760  000012  000001                            MOVB    #RSMSIZ,1(R0)   ;THIS BIG
  (1)  020714  112760  000002  000002                            MOVB    #RSSRD,2(R0)    ;OP CODE IS READ
  (1)  020722  016560  000064  000012                            MOV     REC(R5),10.(R0) ;THIS RECORD
  (1)  020730  116560  000060  000004                            MOVB    DR(R5),4.(R0)   ;THIS DRIVE
  (1)  020736  112760  000000  000003                            MOVB    #0,3.(R0)       ;VERIFY
  (1)  020744  012760  001000  000010                            MOV     #512.,8.(R0)    ;TOTAL BYTES TO READ
  (1)  020752  112760  000020  000005                            MOVB    #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,XSPKNM(R5)   ;SAVE # PACKETS TO EXPEC
  (1)  021052  012723  000002                                    MOV     #RSEND,(R3)+    ;EXPECT AN END ALSO...
```

```
   (1)  021056  012713  000016                                          MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES
   (1)  021062  004737  006600                                          CALL    RSVP            ;SEND
   (1)                                                                                         ;AND RETURN TO SCHEDULER
   (1)
   (1)  021066  032715  002010                                  81$:    BIT     #BIT10!BIT3,@R5 ;RETRY?
   (1)  021072  001500                                          BEQ     79$             ;NO.
   (1)  021074                                                  TURTRY  REC(R5),#512.,DR(R5)    ;YES
   (2)
   (2)
   (2)  021074  012700  027746                                  86$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
   (2)  021100  112710  000002                                          MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
   (2)  021104  112760  000012  000001                                  MOVB    #RSMSIZ,1(R0)   ;THIS BIG
   (2)  021112  112760  000002  000002                                  MOVB    #RSSRD,2(R0)    ;OP CODE-READ
   (2)  021120  016560  000064  000012                                  MOV     REC(R5),10.(R0) ;THIS RECORD
   (2)  021126  116560  000060  000004                                  MOVB    DR(R5),4.(R0)   ;THIS DRIVE
   (2)  021134  105060  000003                                          CLRB    3(R0)           ;PRESET NORM THRESHOLD
   (2)  021140  105715                                                  TSTB    @R5             ;REDUCED?
   (2)  021142  100002                                                  BPL     87$             ;NO
   (2)  021144  105260  000003                                          INCB    3(R0)           ;YES-CHANGE THRESHOLD
   (2)  021150  012760  001000  000010                          87$:    MOV     #512.,8.(R0)    ;# BYTES DESIRED
   (2)  021156  112760  000020  000005                                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE
   (2)  021164  005060  000006                                          CLR     6.(R0)          ;NO SEQUENCE #
   (2)  021170  012701  000012                                          MOV     #RSMSIZ,R1      ;SIZE OF PACKET
   (2)  021174  005721                                                  TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
   (2)  021176  012765  000016  000070                                  MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND
   (2)
   (2)  021204  004737  013674                                          CALL    CHKSUM          ;FORM CHECKSUM R1+COUNT
   (2)  021210  010110                                                  MOV     R1,(R0)         ;INSERT IN PACKET
   (2)
   (2)  021212  012701  001000                                          MOV     #512.,R1                ;SET EXPECTATION
   (2)                                                                                         ;CALC # OF DATA PACKETS
   (2)  021216  012703  000034                                          MOV     #XSFLG,R3       ;OFFSET OF FLAG
   (2)  021222  060503                                                  ADD     R5,R3           ;ABS. ADDR. OF XSFLG
   (2)  021224  005002                                                  CLR     R2              ;PRESET
   (2)  021226  005202                                          83$:    INC     R2              ;# PACKETS EXPECTED
   (2)  021230  012723  000001                                          MOV     #RSDATA,(R3)+   ;LOAD XSFLG
   (2)  021234  012723  000204                                          MOV     #132.,(R3)+     ;AND EXPECT COUNT
   (2)  021240  162701  000200                                          SUB     #128.,R1        ;NEG RESULT LAST TIME
   (2)  021244  101401                                                  BLOS    85$             ;LAST TIME!
   (2)  021246  000767                                                  BR      83$             ;MORE TO DO
   (2)  021250  005202                                          85$:    INC     R2              ;ADD ONE FOR END PACK
   (2)  021252  010265  000032                                          MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPEC
   (2)  021256  012723  000002                                          MOV     #RSEND,(R3)+    ;EXPECT AN END
   (2)  021262  012713  000016                                          MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES
   (2)
   (2)  021266  004737  006600                                          CALL    RSVP            ;SEND
   (2)                                                                                         ;AND RETURN TO SCHEDULER
   (2)
  6946  021276  062704  000002                  ADD     #2,R4           ;POINT TO NEXT DATA
  6947  021302  005764  022766                  TST     TST3PT(R4)      ;END?
  6948  021306  001402                          BEQ     2$              ;YES
  6949  021310  000137  020064                  JMP     1$              ;NO-WRITE, READ NEW DATA
  6950  021314  005004                  2$:     CLR     R4              ;POINT TO FIRST DATA
  6951  021316  062765  000200  000064          ADD     #200,REC(R5)    ;BUT NOW USE ADJACENT RECORD
  6952  021324  032765  001000  000064          BIT     #1000,REC(R5)   ;ALL ADJACENT RECORDS DONE?
  6953  021332  001002                          BNE     3$              ;YES
  6954  021334  000137  020064                  JMP     1$              ;NO-WRITE, READ AT NEW RECORD
```

```
 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$:     INC     DONE            ;YES-SET FLAG
 6961  021372  000207                                  RETURN
 6962  021374                                          ENDTST
  (3)  021374                                                                          L10016:
  (3)  021374  104401                                                                          TRAP    C$ETST
 6963
```

```
 6965                                           .SBTTL  TEST 4 / HIGH ACTIVITY WRITE/READ (128 BYTE/BLOCK MODE)
 6966
 6967                                           ; WRITE THEN READ VARYING DATA FOR ALL PHYSICALLY ADJACENT BLOCKS AROUND
 6968                                           ; A RECORD, GO HALF-WAY INTO REMAINING TAPE REPEAT UNTIL EOT.
 6969
 6970  021376                                           BGNTST
  (3)  021376                                                                                    T4::
 6971  021376                                           TSTID   #TST4
  (1)  021376  012737  021442  003330                                     MOV     #TST4,TSTTOP    ;SAVE ADDR OF TEST
  (1)  021404  004737  006024                             CALL    SETUP           ;INIT UNITS TSTPC
  (1)  021410  004737  005652                             CALL    SETDR           ;GET 1ST DRVS.
  (1)  021414  004737  006072                             CALL    RUN             ;DO TEST
  (1)  021420  004737  005532                             CALL    SWAPDR          ;GET NEXT DRVS.
  (1)  021424  103004                                     BCC     64$             ;BR NO 2ND DRVS
  (1)  021426  004737  006024                             CALL    SETUP           ;REINIT UNITS TSTPC
  (1)  021432  004737  006072                             CALL    RUN             ;REPEAT TEST
  (1)  021436                                                     64$:            ;DONE
 6972  021436                                           EXIT TST
  (3)  021436  104432                                                                    TRAP    C$EXIT
  (3)  021440  001340                                                                    .WORD   L10017 .
 6973
 6974
 6975  021442  012765  000400  000066          TST4:    MOV     #400,TMP(R5)    ;INIT TO HALF OF REMAINING
 6976  021450  005004                                    CLR     R4              ;FOR INDEX INTO DATA TABLE
 6977  021452  005065  000064                           CLR     REC(R5)         ;START AT RECORD 0
 6978  021456  016465  022766  000072          1$:      MOV     TST3PT(R4),PATTEN(R5)   ;GET DATA
 6979  021464                                           TUWRIT  PATTEN(R5),REC(R5),#128.,DR(R5),#BIT7
  (1)  021464  012700  027746                                             72$:    MOV     #TRBUF,R0       ;MAKE COMMAND PACKET:
  (1)  021470  112710  000002                                             MOVB    #RSCMND,@R0     ;COMMAND FLAG
  (1)  021474  112760  000012  000001                                     MOVB    #RSMSIZ,1(R0)   ;THIS SIZE
  (1)  021502  112760  000003  000002                                     MOVB    #RSSWR,2(R0)    ;INSERT OP CODE-WRITE
  (1)  021510  112760  000200  000003                                     MOVB    #BIT7,3.(R0)    ;VERIFY (1 OR 0)
  (1)  021516  116560  000060  000004                                     MOVB    DR(R5),4.(R0)   ;DRIVE #
  (1)  021524  112760  000020  000005                                     MOVB    #020,5.(R0)     ;MAINTENANCE MODE SWITCH
  (1)  021532  005060  000006                                             CLR     6.(R0)          ;NO SEQUENCE #
  (1)  021536  012760  000200  000010                                     MOV     #128.,8.(R0)    ;TOTAL COUNT TO WRITE
  (1)  021544  016560  000064  000012                                     MOV     REC(R5),10.(R0) ;AT  RECORD N
  (1)  021552  012701  000012                                             MOV     #RSMSIZ,R1      ;THE PACKET SIZE PLUS+2
  (1)  021556  005721                                                     TST     (R1)+           ;(FLAG AND COUNT) INTO R
  (1)  021560  012765  000016  000070                                     MOV     #RSSNSZ,SNDCNT(R5)  ;LOAD THE  SIZE TO S
  (1)  021566  004737  013674                                             CALL    CHKSUM          ;R0 --> R1=COUNT
  (1)  021572  010110                                                     MOV     R1,(R0)         ;PUT CHKSUM IN PACKET
  (1)                                                                             ;SET UP EXPECTATIONS:
  (1)  021574  012765  000020  000034                                     MOV     #RSCONT,XSFLG(R5)   ;THE FLAG
  (1)  021602  012765  000001  000036                                     MOV     #1,XSCNT(R5)    ;THE COUNT
  (1)  021610  012765  000001  000032                                     MOV     #1,XSPKNM(R5)   ;THE # PACKETS EXPECTED
  (1)  021616  012702  000200                                             MOV     #128.,R2                ;GET # OF DATA B
  (1)  021622  004737  006600                                             CALL    RSVP            ;SEND (AND RETURN TO SCH
  (1)  021626  032715  000010                                             BIT     #BIT3,@R5       ;FLAG BYTE ERROR?
  (1)  021632  001314                                                     BNE     72$             ;YES
  (1)  021634  042715  010000                                             BIC     #BIT12,@R5      ;FLAG FOR LAST PACKET
  (1)  021640  012700  027746                                     64$:    MOV     #TRBUF,R0       ;POINT TO TOP OF BUFFER
  (1)  021644  020227  000200                                             CMP     R2,#128.        ;START DATA PACKET(S)
  (1)  021650  101004                                                     BHI     65$             ;#128. > 128.!
  (1)  021652  010201                                                     MOV     R2,R1           ;#128.<128.
  (1)  021654  052715  010000                                             BIS     #BIT12,@R5      ;SO LAST PACKET NOW
```

```
(1)  021660  000402                                          BR      66$             ;USE REMAINING COUNT
(1)  021662  012701  000200                            65$:  MOV     #128.,R1        ;USE 128. BYTES
(1)  021666  110160  000001                            66$:  MOVB    R1,1(R0)        ;COPY COUNT TO BUFFER
(1)  021672  010103                                          MOV     R1,R3           ;R3=COUNTER TO LOAD BUFF
(1)  021674  112710  000001                                  MOVB    @RSDATA,@R0     ;FLAG FIRST
(1)  021700  005720                                          TST     (R0)+           ;SKIP COUNT
(1)  021702  116520  000072                            67$:  MOVB    PATTEN(R5),(R0)+        ;INSERT DATA
(1)  021706  005303                                          DEC     R3              ;MORE?
(1)  021710  101374                                          BHI     67$             ;YES
(1)  021712  012700  027746                                  MOV     #TRBUF,R0       ;-->TOP AGAIN
(1)  021716  116001  000001                                  MOVB    1(R0),R1        ;GET COUNT
(1)  021722  042701  177400                                  BIC     #177400,R1      ;ZERO SIGN EXTEND
(1)  021726  010165  000070                                  MOV     R1,SNDCNT(R5)   ;HOW MANY TO SEND PLUS
(1)  021732  062765  000004  000070                          ADD     #4,SNDCNT(R5)   ;FLAG,COUNT,CHKSUM
(1)  021740  062701  000002                                  ADD     #2,R1           ;COMPENSATE FOR FLAG + C
(1)  021744  004737  013674                                  CALL    CHKSUM          ;FOR CHECKSUM CALC.
(1)  021750  110120                                          MOVB    R1,(R0)+        ;CHKSUM INTO PACKET
(1)  021752  000301                                          SWAB    R1              ;EVEN ON AN ODD
(1)  021754  110120                                          MOVB    R1,(R0)+        ;BYTE BOUNDARY
(1)  021756  032715  010000                                  BIT     #BIT12,@R5      ;LAST DATA PACKET?
(1)  021762  001412                                          BEQ     68$             ;NO
(1)  021764  012765  000002  000034                          MOV     #RSEND,XSFLG(R5)  ;YES-EXPECT 'END'
(1)  021772  012765  000016  000036                          MOV     #RSNDSZ,XSCNT(R5)  ;OF THIS SIZE
(1)  022000  012765  000001  000032                          MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
(1)  022006  000411                                          BR      69$             ;SEND
(1)  022010  012765  000020  000034                    68$:  MOV     #RSCONT,XSFLG(R5)  ;(NOT LAST), EXPECT '
(1)  022016  012765  000001  000036                          MOV     #1,XSCNT(R5)    ;AND 1 BYTE
(1)  022024  012765  000001  000032                          MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
(1)  022032  004737  006600                            69$:  CALL    RSVP            ;SEND PACKET
(1)                                                          ;AND RETURN TO SCHEDULER
(1)  022036  032715  000010                                  BIT     #BIT3,@R5       ;FLAG BYTE RETRY?
(1)  022042  001210                                          BNE     72$             ;YES
(1)  022044  032715  002000                                  BIT     #BIT10,@R5      ;RETRY DATA ERROR?
(1)  022050  001004                                          BNE     70$             ;YES
(1)  022052  162702  000200                                  SUB     #128.,R2        ;NO, MORE DATA TO SEND?
(1)  022056  101270                                          BHI     64$             ;YES
(1)  022060  000502                                          BR      71$             ;NO
(1)  022062                                            70$:  TURTRY  REC(R5),#128.,DR(R5)    ;RETRY HERE
(2)
(2)
(2)  022062  012700  027746                            76$:  MOV     #TRBUF,R0       ;FORM CMND PACK:
(2)  022066  112710  000002                                  MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
(2)  022072  112760  000012  000001                          MOVB    #RSMSIZ,1(R0)   ;THIS BIG
(2)  022100  112760  000002  000002                          MOVB    #RSSRD,2(R0)    ;OP CODE-READ
(2)  022106  016560  000064  000012                          MOV     REC(R5),10.(R0) ;THIS RECORD
(2)  022114  116560  000060  000004                          MOVB    DR(R5),4.(R0)   ;THIS DRIVE
(2)  022122  105060  000003                                  CLRB    3(R0)           ;PRESET NORM THRESHOLD
(2)  022126  105715                                          TSTB    @R5             ;REDUCED?
(2)  022130  100002                                          BPL     77$             ;NO
(2)  022132  105260  000003                                  INCB    3(R0)           ;YES-CHANGE THRESHOLD
(2)  022136  012760  000200  000010                    77$:  MOV     #128.,8.(R0)    ;# BYTES DESIRED
(2)  022144  112760  000020  000005                          MOVB    #020,5.(R0)     ;MAINTENANCE MODE
(2)  022152  005060  000006                                  CLR     6.(R0)          ;NO SEQUENCE #
(2)  022156  012701  000012                                  MOV     #RSMSIZ,R1      ;SIZE OF PACKET
(2)  022162  005721                                          TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
(2)  022164  012765  000016  000070                          MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND
```

```
   (2)
   (2)  022172  004737  013674                                  CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
   (2)  022176  010110                                          MOV     R1,(R0)         ;INSERT IN PACKET
   (2)
   (2)  022200  012701  000200                                  MOV     #128.,R1                ;SET EXPECTATION
   (2)                                                                                  ;CALC # OF DATA PACKETS
   (2)  022204  012703  000034                                  MOV     #XSFLG,R3       ;OFFSET OF FLAG
   (2)  022210  060503                                          ADD     R5,R3           ;ABS. ADDR. OF XSFLG
   (2)  022212  005002                                          CLR     R2              ;PRESET
   (2)  022214  005202                                  73$:    INC     R2              ;# PACKETS EXPECTED
   (2)  022216  012723  000001                                  MOV     #RSDATA,(R3)+   ;LOAD XSFLG
   (2)  022222  012723  000204                                  MOV     #132.,(R3)+     ;AND EXPECT COUNT
   (2)  022226  162701  000200                                  SUB     #128.,R1        ;NEG RESULT LAST TIME
   (2)  022232  101401                                          BLOS    75$             ;LAST TIME!
   (2)  022234  000767                                          BR      73$             ;MORE TO DO
   (2)  022236  005202                                  75$:    INC     R2              ;ADD ONE FOR END PACK
   (2)  022240  010265  000032                                  MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPEC
   (2)  022244  012723  000002                                  MOV     #RSEND,(R3)+    ;EXPECT AN END
   (2)  022250  012713  000016                                  MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES
   (2)
   (2)  022254  004737  006600                                  CALL    RSVP            ;SEND
   (2)                                                                                  ;AND RETURN TO SCHEDULER
   (2)
  6980  022270                        TUREAD  REC(R5),#128.,DR(R5),#BIT7
   (1)
   (1)
   (1)  022270  012700  027746                          82$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
   (1)  022274  112710  000002                                  MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
   (1)  022300  112760  000012  000001                          MOVB    #RSMSIZ,1(R0)   ;THIS BIG
   (1)  022306  112760  000002  000002                          MOVB    #RSSRD,2(R0)    ;OP CODE IS READ
   (1)  022314  016560  000064  000012                          MOV     REC(R5),10.(R0) ;THIS RECORD
   (1)  022322  116560  000060  000004                          MOVB    DR(R5),4.(R0)   ;THIS DRIVE
   (1)  022330  112760  000200  000003                          MOVB    #BIT7,3.(R0)    ;VERIFY
   (1)  022336  012760  000200  000010                          MOV     #128.,8.(R0)    ;TOTAL BYTES TO READ
   (1)  022344  112760  000020  000005                          MOVB    #020,5.(R0)     ;MAINTENANCE MODE
   (1)  022352  005060  000006                                  CLR     6.(R0)          ;NO SEQUENCE #
   (1)  022356  012701  000012                                  MOV     #RSMSIZ,R1      ;GET SIZE OF PACKET
   (1)  022362  005721                                          TST     (R1)+           ;+2 FOR CHECKSUM
   (1)  022364  012765  000016  000070                          MOV     #RSSNSZ,SNDCNT(R5)  ;SIZE TO SEND
   (1)  022372  004737  013674                                  CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
   (1)  022376  010110                                          MOV     R1,(R0)         ;INSERT CHECKSUM
   (1)
   (1)  022400  012701  000200                                  MOV     #128.,R1                ;SET EXPECTATION
   (1)                                                                                  ;CALC # OF DATA PACKETS
   (1)  022404  012703  000034                                  MOV     #XSFLG,R3       ;GET OFFSET
   (1)  022410  060503                                          ADD     R5,R3           ;ABS. ADDR. OF XSFLG
   (1)  022412  005002                                          CLR     R2              ;PRESET AS NONE
   (1)  022414  005202                                  78$:    INC     R2              ;# PACKETS EXPECTED
   (1)  022416  012723  000001                                  MOV     #RSDATA,(R3)+   ;LOAD XSFLG
   (1)  022422  012723  000204                                  MOV     #132.,(R3)+     ;AND EXPECTED COUNT
   (1)  022426  162701  000200                                  SUB     #128.,R1        ;NEG RESULT LAST TIME
   (1)  022432  101401                                          BLOS    80$             ;LAST TIME
   (1)  022434  000767                                          BR      78$             ;MORE TO DO
   (1)  022436  005202                                  80$:    INC     R2              ;ADD ONE FOR END PACK
   (1)  022440  010265  000032                                  MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPEC
   (1)  022444  012723  000002                                  MOV     #RSEND,(R3)+    ;EXPECT AN END ALSO...
```

```
  (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                                  81$:    BIT     #BIT10!BIT3,@R5  ;RETRY?
  (1)  022464  001500                                                  BEQ     79$              ;NO.
  (1)  022466                                                          TURTRY  REC(R5),#128.,DR(R5)     ;YES
  (2)
  (2)
  (2)  022466  012700  027746                                  86$:    MOV     #TRBUF,R0        ;FORM CMND PACK:
  (2)  022472  112710  000002                                          MOVB    #RSCMND,@R0      ;MESSAGE PACK TYPE
  (2)  022476  112760  000012  000001                                 MOVB    #RSMSIZ,1(R0)    ;THIS BIG
  (2)  022504  112760  000002  000002                                 MOVB    #RSSRD,2(R0)     ;OP CODE-READ
  (2)  022512  016560  000064  000012                                 MOV     REC(R5),10.(R0)  ;THIS RECORD
  (2)  022520  116560  000060  000004                                 MOVB    DR(R5),4.(R0)    ;THIS DRIVE
  (2)  022526  105060  000003                                          CLRB    3(R0)            ;PRESET NORM THRESHOLD
  (2)  022532  105715                                                  TSTB    @R5              ;REDUCED?
  (2)  022534  100002                                                  BPL     87$              ;NO
  (2)  022536  105260  000003                                          INCB    3(R0)            ;YES-CHANGE THRESHOLD
  (2)  022542  012760  000200  000010                         87$:    MOV     #128.,8.(R0)     ;# BYTES DESIRED
  (2)  022550  112760  000020  000005                                 MOVB    #020,5.(R0)      ;MAINTENANCE MODE
  (2)  022556  005060  000006                                          CLR     6.(R0)           ;NO SEQUENCE #
  (2)  022562  012701  000012                                          MOV     #RSMSIZ,R1       ;SIZE OF PACKET
  (2)  022566  005721                                                  TST     (R1)+            ;PLUS FLAG+COUNT INTO R1
  (2)  022570  012765  000016  000070                                 MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND
  (2)
  (2)  022576  004737  013674                                          CALL    CHKSUM           ;FORM CHECKSUM R1=COUNT
  (2)  022602  010110                                                  MOV     R1,(R0)          ;INSERT IN PACKET
  (2)
  (2)  022604  012701  000200                                          MOV     #128.,R1                 ;SET EXPECTATION
  (2)                                                                                           ;CALC # OF DATA PACKETS
  (2)  022610  012703  000034                                          MOV     #XSFLG,R3        ;OFFSET OF FLAG
  (2)  022614  060503                                                  ADD     R5,R3            ;ABS. ADDR. OF XSFLG
  (2)  022616  005002                                                  CLR     R2               ;PRESET
  (2)  022620  005202                                          83$:    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    85$              ;LAST TIME!
  (2)  022640  000767                                                  BR      83$              ;MORE TO DO
  (2)  022642  005202                                          85$:    INC     R2               ;ADD ONE FOR END PACK
  (2)  022644  010265  000032                                          MOV     R2,XSPKNM(R5)    ;SAVE # PACKETS TO EXPEC
  (2)  022650  012723  000002                                          MOV     #RSEND,(R3)+     ;EXPECT AN END
  (2)  022654  012713  000016                                          MOV     #RSNDSZ,(R3)     ;THIS BIG-14. BYTES
  (2)
  (2)  022660  004737  006600                                          CALL    RSVP             ;SEND
  (2)                                                                                           ;AND RETURN TO SCHEDULER
  (2)
6981   022670  062704  000002                                  ADD     #2,R4            ;POINT TO NEXT DATA
6982   022674  005764  022766                                  TST     TST3PT(R4)       ;END?
6983   022700  001402                                          BEQ     2$               ;YES
6984   022702  000137  021456                                  JMP     1$               ;NO-WRITE, READ NEW DATA
6985   022706  005004                                  2$:     CLR     R4               ;POINT TO FIRST DATA
6986   022710  062765  001000  000064                         ADD     #1000,REC(R5)    ;BUT NOW USE ADJACENT RECORD
6987   022716  032765  004000  000064                         BIT     #4000,REC(R5)    ;ALL ADJACENT RECORDS DONE?
6988   022724  001002                                          BNE     3$               ;YES
6989   022726  000137  021456                                  JMP     1$               ;NO-WRITE, READ AT NEW RECORD
```

```
 6990  022732  162765  004000  000064          3$:     SUB     #4000,REC(R5)    ;RESTORE TO NEXT RECORD
 6991  022740  066565  000066  000064                  ADD     TMP(R5),REC(R5)  ;HALF INTO REST OF TAPE
 6992  022746  006265  000066                          ASR     TMP(R5)          ;HALF OF HALF FOR NEXT TIME
 6993  022752  103402                                  BCS     4$               ;DONE?
 6994  022754  000137  021456                          JMP     1$               ;NO
 6995  022760  005237  003324                  4$:     INC     DONE             ;YES-SET FLAG
 6996  022764  000207                                  RETURN
 6997  022766  000000                          TST3PT: .WORD   000000
 6998  022770  125252                                  .WORD   125252
 6999  022772  177777                                  .WORD   177777
 7000  022774  052525                                  .WORD   052525
 7001  022776  000000                                  .WORD   000000
 7002
 7003
 7004  023000                                          ENDTST
  (3)  023000                                                                  L10017:
  (3)  023000  104401                                                                  TRAP    C$ETST
```

```
 7007
 7008                                   .SBTTL  TEST 5 / WRITE SELECTED NUMBER OF BLOCKS
 7009
 7010  023002                                   BGNTST
  (3)  023002                                                                          T5::
 7011  023002                                   TSTID   #TST5
  (1)  023002  012737  023046  003330                        MOV     #TST5,TSTTOP    ;SAVE ADDR OF TEST
  (1)  023010  004737  006024                                CALL    SETUP           ;INIT UNITS TSTPC
  (1)  023014  004737  005652                                CALL    SETDR           ;GET 1ST DRVS.
  (1)  023020  004737  006072                                CALL    RUN             ;DO TEST
  (1)  023024  004737  005532                                CALL    SWAPDR          ;GET NEXT DRVS.
  (1)  023030  103004                                        BCC     64$             ;BR NO 2ND DRVS
  (1)  023032  004737  006024                                CALL    SETUP           ;REINIT UNITS TSTPC
  (1)  023036  004737  006072                                CALL    RUN             ;REPEAT TEST
  (1)  023042                                          64$:                          ;DONE
 7012  023042                                   EXIT TST
  (3)  023042  104432                                                                          TRAP    C$EXIT
  (3)  023044  000724                                                                          .WORD   L10020-.
 7013
 7014
 7015  023046  005065  000064                TST5:   CLR     REC(R5)         ;START AT REC 0
 7016  023052  013765  003312  000066                MOV     TAPLEN,TMP(R5)  ;GET THE # OF BLOCKS PER TRACK
 7017  023060  005065  000062                        CLR     TRK(R5)         ;TRK(R5)=1ST OR 2ND PASS COUNTER
 7018  023064  016565  000064  000072        1$:     MOV     REC(R5),PATTEN(R5)  ;USE RECORD NO. FOR DATA
 7019  023072  005737  002220                        TST     DRVCHK                  ;ADD DR #?
 7020  023076  001403                                BEQ     10$                     ;NO
 7021  023100  066565  000060  000072                ADD     DR(R5),PATTEN(R5)       ;YES, ADD DRIVE ID
 7022  023106                                10$:    TUWRIT  PATTEN(R5),REC(R5),#512.,DR(R5),#0
  (1)  023106  012700  027746                                        72$:    MOV     #TRBUF,R0       ;MAKE COMMAND PACKET:
  (1)  023112  112710  000002                                        MOVB    #RSCMND,@R0     ;COMMAND FLAG
  (1)  023116  112760  000012  000001                                MOVB    #RSMSIZ,1(R0)   ;THIS SIZE
  (1)  023124  112760  000003  000002                                MOVB    #RSSWR,2(R0)    ;INSERT OP CODE-WRITE
  (1)  023132  112760  000000  000003                                MOVB    #0,3.(R0)       ;VERIFY (1 OR 0)
  (1)  023140  116560  000060  000004                                MOVB    DR(R5),4.(R0)   ;DRIVE #
  (1)  023146  112760  000020  000005                                MOVB    #020,5.(R0)     ;MAINTENANCE MODE SWITCH
  (1)  023154  005060  000006                                        CLR     6.(R0)          ;NO SEQUENCE #
  (1)  023160  012760  001000  000010                                MOV     #512.,8.(R0)    ;TOTAL COUNT TO WRITE
  (1)  023166  016560  000064  000012                                MOV     REC(R5),10.(R0) ;AT  RECORD N
  (1)  023174  012701  000012                                        MOV     #RSMSIZ,R1      ;THE PACKET SIZE PLUS+2
  (1)  023200  005721                                                TST     (R1)+           ;(FLAG AND COUNT) INTO R
  (1)  023202  012765  000016  000070                                MOV     #RSSNSZ,SNDCNT(R5)  ;LOAD THE  SIZE TO S
  (1)  023210  004737  013674                                        CALL    CHKSUM          ;R0 --> R1=COUNT
  (1)  023214  010110                                                MOV     R1,(R0)         ;PUT CHKSUM IN PACKET
  (1)                                                                                        ;SET UP EXPECTATIONS:
  (1)  023216  012765  000020  000034                                MOV     #RSCONT,XSFLG(R5)   ;THE FLAG
  (1)  023224  012765  000001  000036                                MOV     #1,XSCNT(R5)    ;THE COUNT
  (1)  023232  012765  000001  000032                                MOV     #1,XSPKNM(R5)   ;THE # PACKETS EXPECTED
  (1)  023240  012702  001000                                        MOV     #512.,R2                ;GET # OF DATA B
  (1)  023244  004737  006600                                        CALL    RSVP            ;SEND (AND RETURN TO SCH
  (1)  023250  032715  000010                                        BIT     #BIT3,@R5       ;FLAG BYTE ERROR?
  (1)  023254  001314                                                BNE     72$             ;YES
  (1)  023256  042715  010000                                        BIC     #BIT12,@R5      ;FLAG FOR LAST PACKET
  (1)  023262  012700  027746                                64$:    MOV     #TRBUF,R0       ;POINT TO TOP OF BUFFER
  (1)  023266  020227  000200                                        CMP     R2,#128.        ;START DATA PACKET(S)
  (1)  023272  101004                                                BHI     65$             ;#512. > 128.!
  (1)  023274  010201                                                MOV     R2,R1           ;#512.<128.
```

```
(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$:    MOVB    PATTEN(R5),(R0)+        ;INSERT DATA
(1)  023330  005303                                  DEC     R3              ;MORE?
(1)  023332  101374                                  BHI     67$             ;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,SNDCNT(R5)   ;HOW MANY TO SEND PLUS
(1)  023354  062765  000004  000070                  ADD     #4,SNDCNT(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                                  BEQ     68$             ;NO
(1)  023406  012765  000002  000034                  MOV     #RSEND,XSFLG(R5)  ;YES-EXPECT 'END'
(1)  023414  012765  000016  000036                  MOV     #RSNDSZ,XSCNT(R5)  ;OF THIS SIZE
(1)  023422  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
(1)  023430  000411                                  BR      69$             ;SEND
(1)  023432  012765  000020  000034          68$:    MOV     #RSCONT,XSFLG(R5)  ;(NOT LAST), EXPECT
(1)  023440  012765  000001  000036                  MOV     #1,XSCNT(R5)    ;AND 1 BYTE
(1)  023446  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
(1)  023454  004737  006600                  69$:    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    #RSCMND,@R0     ;MESSAGE PACK TYPE
(2)  023514  112760  000012  000001                  MOVB    #RSMSIZ,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     #RSMSIZ,R1      ;SIZE OF PACKET
(2)  023604  005721                                  TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
```

```
   (2) 023606  012765  000016  000070                                    MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND
   (2)
   (2) 023614  004737  013674                                            CALL    CHKSUM              ;FORM CHECKSUM R1=COUNT
   (2) 023620  010110                                                    MOV     R1,(R0)             ;INSERT IN PACKET
   (2)
   (2) 023622  012701  001000                                            MOV     #512.,R1                    ;SET EXPECTATION
   (2)                                                                                               ;CALC # OF DATA PACKETS
   (2) 023626  012703  000034                                            MOV     #XSFLG,R3           ;OFFSET OF FLAG
   (2) 023632  060503                                                    ADD     R5,R3               ;ABS. ADDR. OF XSFLG
   (2) 023634  005002                                                    CLR     R2                  ;PRESET
   (2) 023636  005202                                            73$:    INC     R2                  ;# PACKETS EXPECTED
   (2) 023640  012723  000001                                            MOV     #RSDATA,(R3)+       ;LOAD XSFLG
   (2) 023644  012723  000204                                            MOV     #132.,(R3)+         ;AND EXPECT COUNT
   (2) 023650  162701  000200                                            SUB     #128.,R1            ;NEG RESULT LAST TIME
   (2) 023654  101401                                                    BLOS    75$                 ;LAST TIME!
   (2) 023656  000767                                                    BR      73$                 ;MORE TO DO
   (2) 023660  005202                                            75$:    INC     R2                  ;ADD ONE FOR END PACK
   (2) 023662  010265  000032                                            MOV     R2,XSPKNM(R5)       ;SAVE # PACKETS TO EXPEC
   (2) 023666  012723  000002                                            MOV     #RSEND,(R3)+        ;EXPECT AN END
   (2) 023672  012713  000016                                            MOV     #RSNDSZ,(R3)        ;THIS BIG-14. BYTES
   (2)
   (2) 023676  004737  006600                                            CALL    RSVP                ;SEND
   (2)                                                                                               ;AND RETURN TO SCHEDULER
   (2)
7023 023712  005365  000066                    DEC     TMP(R5)          ;DO ALL RECORDS FOR THIS TRACK?
7024 023716  001404                            BEQ     2$               ;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$:     TST     TRK(R5)          ;DONE 2 TRACKS?
7028 023734  001012                            BNE     TST5EX           ;YES-EXIT
7029 023736  005265  000062                    INC     TRK(R5)          ;NO-SET FLAG FOR NEXT PASS
7030 023742  013765  003336  000064            MOV     SECREC,REC(R5)   ;GET NEW STARTING BLOCK #
7031 023750  013765  003312  000066            MOV     TAPLEN,TMP(R5)   ;RESET # OF BLOCKS
7032 023756  000137  023064                    JMP     1$               ;AND EXECUTE
7033 023762  005237  003324            TST5EX: INC     DONE             ;DONE
7034 023766  000207                            RETURN                   ;RETURN
7035
7036 023770                                    ENDTST
 (3) 023770                                                                    L10020:
 (3) 023770  104401                                                                    TRAP    C$ETST
```

```
 7039                                          .SBTTL  TEST 6 / READ SELECTED NUMBER OF BLOCKS
 7040
 7041  023772                                  BGNTST
  (3)  023772                                                                                  T6::
 7042  023772                                  TSTID   #TST6
  (1)  023772  012737  024036  003330                          MOV     #TST6,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 TFST
  (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    #RSMSIZ,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     #RSMSIZ,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)                                                                          ;CALC # OF DATA PACKETS
  (1)  024212  012703  000034                          MOV     #XSFLG,R3       ;GET OFFSET
  (1)  024216  060503                                  ADD     R5,R3           ;ABS. ADDR. OF XSFLG
  (1)  024220  005002                                  CLR     R2              ;PRESET AS NONE
  (1)  024222  005202                          64$:    INC     R2              ;# PACKETS EXPECTED
  (1)  024224  012723  000001                          MOV     #RSDATA,(R3)+   ;LOAD XSFLG
  (1)  024230  012723  000204                          MOV     #132.,(R3)+     ;AND EXPECTED COUNT
  (1)  024234  162701  000200                          SUB     #128.,R1        ;NEG RESULT LAST TIME
  (1)  024240  101401                                  BLOS    66$             ;LAST TIME
  (1)  024242  000767                                  BR      64$             ;MORE TO DO
```

```
(1)  024244  005202                                          66$:  INC     R2                ;ADD ONE FOR END PACK
(1)  024246  010265  000032                                       MOV     R2,XSPKNM(R5)     ;SAVE # PACKETS TO EXPEC
(1)  024252  012723  000002                                       MOV     #RSEND,(R3)+      ;EXPECT AN END ALSO...
(1)  024256  012713  000016                                       MOV     #RSNDSZ,(R3)      ;THIS BIG-14. BYTES
(1)  024262  004737  006600                                       CALL    RSVP              ;SEND
(1)                                                               ;AND RETURN TO SCHEDULER
(1)  024266  032715  002010                                  67$:  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$:  MOV     #TRBUF,R0         ;FORM CMND PACK:
(2)  024300  112710  000002                                       MOVB    #RSCMND,@R0       ;MESSAGE PACK TYPE
(2)  024304  112760  000012  000001                              MOVB    #RSMSIZ,1(R0)     ;THIS BIG
(2)  024312  112760  000002  000002                              MOVB    #RSSRD,2(R0)      ;OP CODE-READ
(2)  024320  016560  000064  000012                              MOV     REC(R5),10.(R0)   ;THIS RECORD
(2)  024326  116560  000060  000004                              MOVB    DR(R5),4.(R0)     ;THIS DRIVE
(2)  024334  105060  000003                                       CLRB    3(R0)             ;PRESET NORM THRESHOLD
(2)  024340  105715                                               TSTB    @R5               ;REDUCED?
(2)  024342  100002                                               BPL     73$               ;NO
(2)  024344  105260  000003                                       INCB    3(R0)             ;YES-CHANGE THRESHOLD
(2)  024350  012760  001000  000010                         73$:  MOV     #512.,8.(R0)      ;# BYTES DESIRED
(2)  024356  112760  000020  000005                              MOVB    #020,5.(R0)       ;MAINTENANCE MODE
(2)  024364  005060  000006                                       CLR     6.(R0)            ;NO SEQUENCE #
(2)  024370  012701  000012                                       MOV     #RSMSIZ,R1        ;SIZE OF PACKET
(2)  024374  005721                                               TST     (R1)+             ;PLUS FLAG+COUNT INTO R1
(2)  024376  012765  000016  000070                              MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND
(2)
(2)  024404  004737  013674                                       CALL    CHKSUM            ;FORM CHECKSUM R1=COUNT
(2)  024410  010110                                               MOV     R1,(R0)           ;INSERT IN PACKET
(2)
(2)  024412  012701  001000                                       MOV     #512.,R1                ;SET EXPECTATION
(2)                                                               ;CALC # OF DATA PACKETS
(2)  024416  012703  000034                                       MOV     #XSFLG,R3         ;OFFSET OF FLAG
(2)  024422  060503                                               ADD     R5,R3             ;ABS. ADDR. OF XSFLG
(2)  024424  005002                                               CLR     R2                ;PRESET
(2)  024426  005202                                          69$:  INC     R2                ;# PACKETS EXPECTED
(2)  024430  012723  000001                                       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                                          71$:  INC     R2                ;ADD ONE FOR END PACK
(2)  024452  010265  000032                                       MOV     R2,XSPKNM(R5)     ;SAVE # PACKETS TO EXPEC
(2)  024456  012723  000002                                       MOV     #RSEND,(R3)+      ;EXPECT AN END
(2)  024462  012713  000016                                       MOV     #RSNDSZ,(R3)      ;THIS BIG-14. BYTES
(2)
(2)  024466  004737  006600                                       CALL    RSVP              ;SEND
(2)                                                               ;AND RETURN TO SCHEDULER
(2)
7054  024476  005365  000066                    DEC     TMP(R5)          ;DO ALL RECORDS THIS TRACK?
7055  024502  001404                            BEQ     2$               ;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     SECREC,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          TST6EX:     INC     DONE             ;DONE
 7065  024552  000207                              RETURN                   ;RETURN
 7066
 7067  024554                                      ENDTST
  (3)  024554                                                                          L10021:
  (3)  024554  104401                                                                          TRAP    C$ETS*
```

```
                                .SBTTL  TEST 7 / WRITE VERIFY SELECTED NUMBER OF BLOCKS

                                        BGNTST
                                                                                T7::
                                        TSTID   #TST7
                                                        MOV     #TST7,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
                                                64$:                            ;DONE
                                        EXIT TST
                                                                                        TRAP    C$EXIT
                                                                                        .WORD   L10022 .

                                TST7:   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
                                1$:     MOV     REC(R5),PATTEN(R5)  ;USE RECORD NO. FOR DATA
                                        TST     DRVCHK          ;ADD DR #?
                                        BEQ     10$             ;NO
                                        ADD     DR(R5),PATTEN(R5)       ;ADD DRIVE ID
                                10$:    TUWRIT  PATTEN(R5),REC(R5),#512.,DR(R5),#1
                                                72$:    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    #1,3.(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     #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     #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.
                                                        BIS     #BIT12,@R5      ;SO LAST PACKET NOW
```

```
(1)  025056  000402                                          BR      66$             ;USE REMAINING COUNT
(1)  025060  012701  000200                            65$:  MOV     #128.,R1        ;USE 128. BYTES
(1)  025064  110160  000001                            66$:  MOVB    R1,1(R0)        ;COPY COUNT TO BUFFER
(1)  025070  010103                                          MOV     R1,R3           ;R3=COUNTER TO LOAD BUFF
(1)  025072  112710  000001                                  MOVB    #RSDATA,@R0     ;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     #TRBUF,R0       ;-->TOP AGAIN
(1)  025114  116001  000001                                  MOVB    1(R0),R1        ;GET COUNT
(1)  025120  042701  177400                                  BIC     #177400,R1      ;ZERO SIGN EXTEND
(1)  025124  010165  000070                                  MOV     R1,SNDCNT(R5)   ;HOW MANY TO SEND PLUS
(1)  025130  062765  000004  000070                          ADD     #4,SNDCNT(R5)   ;FLAG,COUNT,CHKSUM
(1)  025136  062701  000002                                  ADD     #2,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     #BIT12,@R5      ;LAST DATA PACKET?
(1)  025160  001412                                          BEQ     68$             ;NO
(1)  025162  012765  000002  000034                          MOV     #RSEND,XSFLG(R5)  ;YES-EXPECT 'END'
(1)  025170  012765  000016  000036                          MOV     #RSNDSZ,XSCNT(R5)  ;OF THIS SIZE
(1)  025176  012765  000001  000032                          MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
(1)  025204  000411                                          BR      69$             ;SEND
(1)  025206  012765  000020  000034                    68$:  MOV     #RSCONT,XSFLG(R5)  ;(NOT LAST), EXPECT '
(1)  025214  012765  000001  000036                          MOV     #1,XSCNT(R5)    ;AND 1 BYTE
(1)  025222  012765  000001  000032                          MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
(1)  025230  004737  006600                            69$:  CALL    RSVP            ;SEND PACKET
(1)                                                          ;AND RETURN TO SCHEDULER
(1)  025234  032715  000010                                  BIT     #BIT3,@R5       ;FLAG BYTE RETRY?
(1)  025240  001210                                          BNE     72$             ;YES
(1)  025242  032715  002000                                  BIT     #BIT10,@R5      ;RETRY DATA ERROR?
(1)  025246  001004                                          BNE     70$             ;YES
(1)  025250  162702  000200                                  SUB     #128.,R2        ;NO, MORE DATA TO SEND?
(1)  025254  101270                                          BHI     64$             ;YES
(1)  025256  000502                                          BR      71$             ;NO
(1)  025260                                            70$:  TURTRY  REC(R5),#512..DR(R5)     ;RETRY HERE
(2)
(2)
(2)  025260  012700  027746                            76$:  MOV     #TRBUF,R0       ;FORM CMND PACK:
(2)  025264  112710  000002                                  MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
(2)  025270  112760  000012  000001                          MOVB    #RSMSIZ,1(R0)   ;THIS BIG
(2)  025276  112760  000002  000002                          MOVB    #RSSRD,2(R0)    ;OP CODE-READ
(2)  025304  016560  000064  000012                          MOV     REC(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    @R5             ;REDUCED?
(2)  025326  100002                                          BPL     77$             ;NO
(2)  025330  105260  000003                                  INCB    3(R0)           ;YES-CHANGE THRESHOLD
(2)  025334  012760  001000  000010                    77$:  MOV     #512.,8.(R0)    ;# BYTES DESIRED
(2)  025342  112760  000020  000005                          MOVB    #020,5.(R0)     ;MAINTENANCE MODE
(2)  025350  005060  000006                                  CLR     6.(R0)          ;NO SEQUENCE #
(2)  025354  012701  000012                                  MOV     #RSMSIZ,R1      ;SIZE OF PACKET
(2)  025360  005721                                          TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
(2)  025362  012765  000016  000070                          MOV     #RSSNSZ,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$:    INC     R2              ;ADD ONE FOR END PACK
 (2)  025436  010265  000032                                          MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPEC
 (2)  025442  012723  000002                                          MOV     #RSEND,(R3)+    ;EXPECT AN END
 (2)  025446  012713  000016                                          MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES
 (2)
 (2)  025452  004737  006600                                          CALL    RSVP            ;SEND
 (2)                                                                                          ;AND RETURN TO SCHEDULER
 (2)
7085  025466  005365  000066                                  DEC     TMP(R5)         ;DO ALL RECORDS FOR THIS TRACK?
7086  025472  001404                                          BEQ     2$              ;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                                                                                  L10022:
 (3)  025544  104401                                                                                  TRAP    C$ETST
```

```
                                .SBTTL  TEST 8 / READ-REDUCED THRESHOLD SELECTED NUMBER OF BLOCKS

                                        BGNTST
                                                                                T8::
                                        TSTID   #TST8
                                                        MOV     #TST8,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
                                                64$:                            ;DONE
                                        EXIT TST
                                                                                        TRAP    C$EXIT
                                                                                        .WORD   L10023-.

                                TST8:   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
                                1$:     MOV     REC(R5),PATTEN(R5)  ;USE RECORD NO. FOR DATA
                                        TST     DRVCHK          ;ADD DR #?
                                        BEQ      10$            ;NO
                                        ADD     DR(R5),PATTEN(R5)       ;ADD DRIVE ID
                                10$:    TUREAD  REC(R5),#512.,DR(R5),#1


                                                        68$:    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(R5),10.(R0) ;THIS RECORD
                                                                MOVB    DR(R5),4.(R0)   ;THIS DRIVE
                                                                MOVB    #1,3.(R0)       ;VERIFY
                                                                MOV     #512.,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     #512.,R1                ;SET EXPECTATION
                                                                                        ;CALC # OF DATA PACKETS
                                                                MOV     #XSFLG,R3       ;GET OFFSET
                                                                ADD     R5,R3           ;ABS. ADDR. OF XSFLG
                                                                CLR     R2              ;PRESET AS NONE
                                                        64$:    INC     R2              ;# PACKETS EXPECTED
                                                                MOV     #RSDATA,(R3)+   ;LOAD XSFLG
                                                                MOV     #132.,(R3)+     ;AND EXPECTED COUNT
                                                                SUB     #128.,R1        ;NEG RESULT LAST TIME
                                                                BLOS    66$             ;LAST TIME
                                                                BR      64$             ;MORE TO DO
```

```
   (1)  026020  005202                                          66$:  INC     R2                ;ADD ONE FOR END PACK
   (1)  026022  010265  000032                                        MOV     R2,XSPKNM(R5)     ;SAVE # PACKETS TO EXPEC
   (1)  026026  012723  000002                                        MOV     #RSEND,(R3)+      ;EXPECT AN END ALSO...
   (1)  026032  012713  000016                                        MOV     #RSNDSZ,(R3)      ;THIS BIG-14. BYTES
   (1)  026036  004737  006600                                        CALL    RSVP              ;SEND
   (1)                                                                                           ;AND RETURN TO SCHEDULER
   (1)  026042  032715  002010                                  67$:  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$:  MOV     #TRBUF,R0         ;FORM CMND PACK:
   (2)  026054  112710  000002                                        MOVB    #RSCMND,@R0       ;MESSAGE PACK TYPE
   (2)  026060  112760  000012  000001                               MOVB    #RSMSIZ,1(R0)     ;THIS BIG
   (2)  026066  112760  000002  000002                               MOVB    #RSSRD,2(R0)      ;OP CODE-READ
   (2)  026074  016560  000064  000012                               MOV     REC(R5),10.(R0)   ;THIS RECORD
   (2)  026102  116560  000060  000004                               MOVB    DR(R5),4.(R0)     ;THIS DRIVE
   (2)  026110  105060  000003                                        CLRB    3(R0)             ;PRESET NORM THRESHOLD
   (2)  026114  105715                                                TSTB    @R5               ;REDUCED?
   (2)  026116  100002                                                BPL     73$               ;NO
   (2)  026120  105260  000003                                        INCB    3(R0)             ;YES-CHANGE THRESHOLD
   (2)  026124  012760  001000  000010                          73$:  MOV     #512.,8.(R0)      ;# BYTES DESIRED
   (2)  026132  112760  000020  000005                               MOVB    #020,5.(R0)       ;MAINTENANCE MODE
   (2)  026140  005060  000006                                        CLR     6.(R0)            ;NO SEQUENCE #
   (2)  026144  012701  000012                                        MOV     #RSMSIZ,R1        ;SIZE OF PACKET
   (2)  026150  005721                                                TST     (R1)+             ;PLUS FLAG+COUNT INTO R1
   (2)  026152  012765  000016  000070                               MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND
   (2)
   (2)  026160  004737  013674                                        CALL    CHKSUM            ;FORM CHECKSUM R1=COUNT
   (2)  026164  010110                                                MOV     R1,(R0)           ;INSERT IN PACKET
   (2)
   (2)  026166  012701  001000                                        MOV     #512.,R1                  ;SET EXPECTATION
   (2)                                                                                           ;CALC # OF DATA PACKETS
   (2)  026172  012703  000034                                        MOV     #XSFLG,R3         ;OFFSET OF FLAG
   (2)  026176  060503                                                ADD     R5,R3             ;ABS. ADDR. OF XSFLG
   (2)  026200  005002                                                CLR     R2                ;PRESET
   (2)  026202  005202                                          69$:  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     #128.,R1          ;NEG RESULT LAST TIME
   (2)  026220  101401                                                BLOS    71$               ;LAST TIME!
   (2)  026222  000767                                                BR      69$               ;MORE TO DO
   (2)  026224  005202                                          71$:  INC     R2                ;ADD ONE FOR END PACK
   (2)  026226  010265  000032                                        MOV     R2,XSPKNM(R5)     ;SAVE # PACKETS TO EXPEC
   (2)  026232  012723  000002                                        MOV     #RSEND,(R3)+      ;EXPECT AN END
   (2)  026236  012713  000016                                        MOV     #RSNDSZ,(R3)      ;THIS BIG-14. BYTES
   (2)
   (2)  026242  004737  006600                                        CALL    RSVP              ;SEND
   (2)                                                                                           ;AND RETURN TO SCHEDULER
   (2)
  7116  026252  005365  000066                DEC     TMP(R5)       ;DO ALL RECORDS THIS TRACK?
  7117  026256  001404                          BEQ     2$            ;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     TST8EX        ;YES-EXIT
```

```
 7122  026276  005265  000062                        INC     TRK(R5)          ;NO-SET FLAG FOR NEXT PASS
 7123  026302  013765  003336  000064                MOV     SECREC,REC(R5)   ;GET NEW STARTING BLOCK #
 7124  026310  013765  003312  000066                MOV     TAPLEN,TMP(R5)   ;RESET # OF BLOCKS
 7125  026316  000137  025630                        JMP     1$               ;AND EXECUTE
 7126  026322  005237  003324              TST8EX:   INC     DONE             ;DONE
 7127  026326  000207                               RETURN                   ;RETURN
 7128
 7129  026330                                       ENDTST
  (3)  026330                                                                          L10023:
  (3)  026330  104401                                                                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST 9 / TESTS MODIFIED RADIAL SERIAL PROTOCOL

                                                BGNTST
 (3) 026332                                                                                     T9::

                                                MOV     #TST9,TSTTOP    ;SAVE ADDR OF TEST
                                                CALL    SETUP           ;INIT UNITS TSTPC
                                                CALL    RUN             ;DO TEST
                                                                        ;DONE

                                                EXIT TST
 (3) 026350  104432                                                                             TRAP    C$EXIT
 (3) 026352  000662                                                                             .WORD   L10024-.

026354  012737  000001  003344        TST9:   MOV     #1,TEST9        ;INDICATES 1ST PART OF TST 8
026362  012700  027746                64$:    MOV     #TRBUF,R0       ;FORM COMMAND PACKET
026366  112710  000002                        MOVB    #RSCMND,@R0     ;COMMAND FLAG
026372  112760  000012  000001              MOVB    #RSMSIZ,1(R0)   ;SIZE OF MESSAGE
026400  112760  000012  000002              MOVB    #RSSGET,2(R0)   ;GET CHARACTERISTICS
026406  105060  000003                      CLRB    3(R0)           ;NO MODIFIER.
026412  005060  000004                      CLR     4(R0)           ;NO DRIVE OR SWITCHES
026416  005060  000006                      CLR     6(R0)           ;NO SEQUENCE NUMBER
026422  005060  000010                      CLR     8.(R0)          ;NO BYTES
026426  005060  000012                      CLR     10.(R0)         ;NO RECORD #
026432  012701  000012                      MOV     #RSMSIZ,R1      ;GET SIZE
026436  005721                            TST     (R1)+           ;+2 FOR CHECKSUM
026440  012765  000016  000070            MOV     #RSSNSZ,SNDCNT(R5)  ;SIZE TO SEND
026446  004737  013674                    CALL    CHKSUM          ;FORM CHECKSUM
026452  010110                            MOV     R1,(R0)         ;INSERT INTO PACKET
026454  012765  000001  000034            MOV     #RSDATA,XSFLG(R5)  ;EXPECT DATA PACKET
026462  012765  000034  000036            MOV     #RSGCDP,XSCNT(R5)  ;THIS BIG
026470  012765  000001  000032            MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
                                                                ;SEND
026476  004737  006600                    CALL    RSVP            ;RETURN TO SCHEDULER

026502  004737  014030                    CALL    DOBRK           ;CLR POTENTIAL INTERFACE ERROR

026506  032715  000010                    BIT     #BIT3,@R5       ;RETRY?(BAD FLAG)
026512  001323                            BNE     64$             ;YES

026514  012737  000002  003344            MOV     #2,TEST9        ;INDICATE 2ND PART OF TST 8

026522  012700  027746                65$:    MOV     #TRBUF,R0       ;-->(POINT TO) XMIT BUFFER
026526  112710  000002                        MOVB    #RSCMND,@R0     ;FORM COMMAND MESSAGE PACK
026532  112760  000012  000001              MOVB    #RSMSIZ,1(R0)   ;THIS BIG
026540  112760  000001  000002              MOVB    #RSSNIT,2(R0)   ;OP CODE IS INITIALIZE
026546  013760  000064  000012              MOV     REC,10.(R0)     ;TO THIS RECORD
026554  105060  000003                      CLRB    3.(R0)          ;NO MODIFIER
026560  105060  000004                      CLRB    4.(R0)          ;NO DRIVE
026564  112760  000010  000005              MOVB    #BIT03,5.(R0)   ;SET MRSP SWITCH
026572  005060  000006                      CLR     6.(R0)          ;NO SEQUENCE #
026576  005060  000010                      CLR     8.(R0)          ;0 BYTE COUNT
026602  012701  000012                      MOV     #RSMSIZ,R1      ;BUFFER COUNT
026606  005721                            TST     (R1)+           ;PLUS FLAG + BCNT
                                                                ;FOR CHECKSUM CALC
026610  004737  013674                    CALL    CHKSUM          ;R0-->TOP   R1=# OF BYTES
```

```
 7185  026614  010110                          MOV     R1,(R0)          ;INSERT INTO PACKET
 7186                                                                     ;SET UP EXPECTATIONS:
 7187  026616  012765  000016  000070          MOV     #RSSNSZ,SNDCNT(R5)  ;HOW MANY TO SEND
 7188  026624  112765  000002  000034          MOVB    #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                  MOVB    #RSCMND,@R0      ;FORM COMMAND MESSAGE PACK
 7200  026670  112760  000012  000001          MOVB    #RSMSIZ,1(R0)    ;THIS BIG
 7201  026676  112760  000000  000002          MOVB    #RSSNOP,2(R0)    ;OP CODE IS NO-OPERATION
 7202  026704  013760  000064  000012          MOV     REC,10.(R0)      ;TO THIS RECORD
 7203  026712  105060  000003                  CLRB    3.(R0)           ;NO MODIFIER
 7204  026716  105060  000004                  CLRB    4.(R0)           ;NO DRIVE
 7205  026722  112760  000010  000005          MOVB    #BIT03,5.(R0)    ;SET MRSP SWITCH
 7206  026730  005060  000006                  CLR     6.(R0)           ;NO SEQUENCE #
 7207  026734  005060  000010                  CLR     8.(R0)           ;NO BYTE COUNT
 7208  026740  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,SNDCNT(R5)  ;HOW MANY TO SEND
 7215  026762  112765  000002  000034          MOVB    #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     ENDT9            ;NO
 7230  027034  005037  027412                  CLR     UNITNO           ;SET UNIT # TO ZERO
 7231  027040                                  PRINTF  #MSAGE1          ;PRINT HEADER
                                                                                  MOV     #MSAGE1,
  (7)  027040  012746  027236                                                    MOV     #1,-(SP)
  (6)  027044  012746  000001                                                    MOV     SP,R0
  (3)  027050  010600                                                            TRAP    C$PNTF
  (4)  027052  104417                                                            ADD     #4,SP
  (4)  027054  062706  000004
 7232  027060  012737  003352  003314          MOV     #BLKTBL,DEVPTR   ;SET ALL UNITS
 7233  027066  017705  154222          1$:     MOV     @DEVPTR,R5       ;GET POINTER
 7234  027072  005765  000000                  TST     STATUS(R5)       ;IS UNIT ABORTED
 7235  027076  100431                          BMI     3$               ;YES
```

```
 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,R0
  (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$:   PRINTF  @MSAGE3,UNITNO    ;MESSAGE FOR NON-MODIFIED UNIT
  (8)  027134  013746  027412                                                      MOV     UNITNO,-
  (7)  027140  012746  027333                                                      MOV     @MSAGE3,
  (6)  027144  012746  000002                                                      MOV     #2,-(SP)
  (3)  027150  010600                                                              MOV     SP,R0
  (4)  027152  104417                                                              TRAP    C$PNTF
  (4)  027154  062706  000006                                                      ADD     #6,SP
 7241  027160  000412                            BR      4$                ;SEE IF LAST UNIT
 7242  027162                              3$:   PRINTF  @MSAGE4,UNITNO    ;MESSAGE FOR ABORTED UNIT
  (8)  027162  013746  027412                                                      MOV     UNITNO,
  (7)  027166  012746  027362                                                      MOV     @MSAGE4,
  (6)  027172  012746  000002                                                      MOV     #2,-(SP)
  (3)  027176  010600                                                              MOV     SP,R0
  (4)  027200  104417                                                              TRAP    C$PNTF
  (4)  027202  062706  000006                                                      ADD     #6,SP
 7243  027206  023727  003314  003370        4$:   CMP     DEVPTR,@LSTDEV    ;IS THIS THE LAST DEVICE
 7244  027214  103006                            BHIS    ENDT9             ;YES
 7245  027216  062737  000002  003314            ADD     #2,DEVPTR         ;GET NEXT UNIT
 7246  027224  005237  027412                    INC     UNITNO            ;INC UNIT #
 7247  027230  000716                            BR      1$
 7248
 7249  027232  000207                      ENDT9:RETURN
 7250
 7251  027234                                    ENDTST
  (3)  027234                                                          L10024:
  (3)  027234  104401                                                              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><O1><S9><S9><ARSP/MRSP!
 7255  027333     045  022516  034523    MSAGE3: .ASCIZ  /<N><S9><S2><O1><S9><S9><ARSP/
 7256  027362  047045  051445  022471    MSAGE4: .ASCIZ  /<N><S9><S2><O1><S9><S9><A---/
 7257          027412                            .EVEN
 7258  027412  000000                    UNITNO: .WORD
```

```
                                                .SBTTL  PATCH AREA
000144                                                  .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
```

```
                                        .TITLE PARAMETER CODING
                                        .SBTTL  HARDWARE PARAMETER CODING SECTION
041364                                          BGNMOD
                                        ;++
                                        ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
                                        ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                        ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                        ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                        ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                        ; WITH THE OPERATOR.
                                        ;--
041364                                          BGNHRD
(3) 041364  000021                                                              .WORD L10025-L$H
(3) 041366                                                              L$HARD::
041366                                          GPRMA   MSG1,0,0,160000,177777,YES
(4) 041366  000031                                                              .WORD   T$CODE
(4) 041370  041430                                                              .WORD   MSG1
(4) 041372  160000                                                              .WORD   T$LOLIM
(4) 041374  177777                                                              .WORD   T$HILIM
041376                                          GPRMA   MSG1B,2,0,0,776,YES
(4) 041376  001031                                                              .WORD   T$CODE
(4) 041400  041441                                                              .WORD   MSG1B
(4) 041402  000000                                                              .WORD   T$LOLIM
(4) 041404  000776                                                              .WORD   T$HILIM
041406                                          GPRML   MSG1C,6,1,YES
(4) 041406  003130                                                              .WORD   T$CODE
(4) 041410  041456                                                              .WORD   MSG1C
(4) 041412  000001                                                              .WORD   1
041414                                          GPRML   MSG2,4,1,YES
(4) 041414  002130                                                              .WORD   T$CODE
(4) 041416  041474                                                              .WORD   MSG2
(4) 041420  000001                                                              .WORD   1
041422                                          GPRML   MSG3,4,2,YES
(4) 041422  002130                                                              .WORD   T$CODE
(4) 041424  041511                                                              .WORD   MSG3
(4) 041426  000002                                                              .WORD   2

041430                                          ENDHRD
(2)                                                                             .EVEN
(3) 041430                                                              L10025:

041430  052524  034065  041440    MSG1:   .ASCIZ  /TU58 CSR/
041441     126  041505  047524    MSG1B:  .ASCIZ  /VECTOR ADDR./
041456  042120  020124  047111    MSG1C:  .ASCIZ  /PDT INTERFACE/
041474  042524  052123  042040    MSG2:   .ASCIZ  /TEST DRIVE 0/
041511     124  051505  020124    MSG3:   .ASCIZ  /TEST DRIVE 1/
                                        .EVEN
```

```
PARAMETER CODING        MACY11 30(1046)  25 JAN 84  08:33  PAGE 62
CZTUUF.P11    25 JAN 84 08:09           SOFTWARE PARAMETER CODING SECTION                                    SEQ 0125


                                        .SBTTL  SOFTWARE PARAMETER CODING SECTION
                                        ;++
                                        ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                        ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                        ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                        ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                        ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                        ; WITH THE OPERATOR.
                                        ;--

041526                                          BGNSFT
(3) 041526  000034                                                                      .WORD L10026-L$S
(3) 041530                                                                      L$SOFT::
041530                                          GPRMD   MSG4,0,D,1777,8.,512.,YES
(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$HILIM
041542                                          GPRML   MSG4B,10.,,YES
(4) 041542  004130                                                                      .WORD   T$CODE
(4) 041544  041665                                                                      .WORD   MSG4B
(4) 041546  000001                                                                      .WORD   1
041550                                          GPRML   MSG5,2,1,YES
(4) 041550  001130                                                                      .WORD   T$CODE
(4) 041552  041727                                                                      .WORD   MSG5
(4) 041554  000001                                                                      .WORD   1
041556                                          GPRML   MSG6,6,1,YES
(4) 041556  003130                                                                      .WORD   T$CODE
(4) 041560  041761                                                                      .WORD   MSG6
(4) 041562  000001                                                                      .WORD   1
041564                                          GPRML   MSG7,4,1,YES
(4) 041564  002130                                                                      .WORD   T$CODE
(4) 041566  042006                                                                      .WORD   MSG7
(4) 041570  000001                                                                      .WORD   1
041572                                          GPRMD   MSG8,10.,D,377,1,254.,YES
(4) 041572  005052                                                                      .WORD   T$CODE
(4) 041574  042034                                                                      .WORD   MSG8
(4) 041576  000377                                                                      .WORD   377
(4) 041600  000001                                                                      .WORD   T$LOLIM
(4) 041602  000376                                                                      .WORD   T$HILIM
041604                                          GPRML   MSG9,12.,1,YES
(4) 041604  006130                                                                      .WORD   T$CODE
(4) 041606  042075                                                                      .WORD   MSG9
(4) 041610  000001                                                                      .WORD   1
041612                                          GPRML   MSG10,14.,1,YES
(4) 041612  007130                                                                      .WORD   T$CODE
(4) 041614  042142                                                                      .WORD   MSG10
(4) 041616  000001                                                                      .WORD   1
041620                                  SFTOUT: ENDSFT
(2)                                                                                     .EVEN
(3) 041620                                                                      L10026:
041620  052516  041115  051105        MSG4:   .ASCIZ  'NUMBER OF BLOCKS:TEST 5-8 (8 TO 512)'
041665     101  042104  042040        MSG4B:  .ASCIZ  /ADD DR # TO DATA PATTERN:TEST 5-8/
041727     123  040524  044524        MSG5:   .ASCIZ  /STATISTICS PRINTED AT EOP/
041761     103  046517  040520        MSG6:   .ASCIZ  /COMPARE DATA ON READ/
```

```
 7441  042006  051120  047111  020124          MSG7:   .ASCIZ  /PRINT PACKET ON ERROR/
 7442  042034  020043  051105  047522          MSG8:   .ASCIZ  /# ERRORS = DVC FATAL IF 'EVL'SET/
 7443  042075     120  044522  052116          MSG9:   .ASCIZ  /PRINT UNIT PROTOCOL SUMMARY (TEST 9)/
 7444  042142  042524  052123  047440          MSG10:  .ASCIZ  /TEST ONLY DRIVE 0 IN TEST 3/
 7445                                                  .EVEN
```

```
 7448           000016                          .REPT   14.              ;LASTAD CORRECTION
 7449                                           .WORD
 7450                                           .ENDR
 7457  042232                                   LASTAD
  (2)
  (2)  042232  042252                                                                .EVEN
  (2)  042234  000006                                                                .WORD T$FREE
  (2)  042236                                                                        .WORD T$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
```

```
ABNDX = 000004 G      4208#  6002*  6015   6017
ABO     013166        6012   6026   6039   6055#
ABOMSG  017300        6761   6791#
ABONM   006552        5123*  5140*  5145   5153#
ADR   = 000020 G      3962#
ALLGON  003342 G      4177#  5149*  6625*  6717
ASSEMB= 000010        3772
BDATA = 000134 G      4260#  6496   6506
BDBYTS  014746        6344*  6355*  6358   6363   6373#
BDCHK = 000022 G      3990#  5708   5765
BDCOM = 000014 G      3987#  5815   6027
BIT0  = 000001 G      3962#  4935   5536   6018   6129   6130   6132   6143   6218
BIT00 = 000001 G      3962#
BIT01 = 000002 G      3962#
BIT02 = 000004 G      3962#
BIT03 = 000010 G      3962#  7178   7205
BIT04 = 000020 G      3962#
BIT05 = 000040 G      3962#
BIT06 = 000100 G      3962#
BIT07 = 000200 G      3962#
BIT08 = 000400 G      3962#
BIT09 = 001000 G      3962#
BIT1  = 000002 G      3962#  5105   5296   5537   5637   5801   5872
BIT10 = 002000 G      3962#  5103   5225   5821   5841   5849   5870   6944   6945   6979   6980   7022   7053
                      7084   7115
BIT11 = 004000 G      3962#  6482   6562   6583   6684
BIT12 = 010000 G      3962#  6944   6979   7022   7084
BIT13 = 020000 G      3962#  5093   5101   5724   5736   5805
BIT14 = 040000 G      3962#  5129   5132   5942   6268   6589
BIT15 = 100000 G      3962#  4933   5112   5338   5360   5403   5422   5442   5473   5645   5807   6561   6608
                      6684   6758
BIT2  = 000004 G      3962#  5100   5114   5296   5538   5637   5801   5803   5872
BIT3  = 000010 G      3962#  5091   5539   5688   5724   5736   6895   6913   6944   6945   6979   6980   7022
                      7053   7084   7115   7166   7195   7222
BIT4  = 000020 G      3962#  5219   5223   5540   6074   6220
BIT5  = 000040 G      3962#  5241   5243   5541
BIT6  = 000100 G      3962#  5542   5812   5873   6245   6253   6260   6276   6277   6289   6297   6364
BIT7  = 000200 G      3962#  5543   5841   5849   5855   5857   5859   5870   6979   6980
BIT8  = 000400 G      3962#  4972   5095   5229   5238   5834   5863   6590
BIT9  = 001000 G      3962#  4937   5213   5818   5845   5928   6592
BLKEND= 000202 G      4280#  6600
BLKER   003334 G      4174#  5210*  6633*
BLKSIZ= 000212 G      4284#  4306   4307   4308   4309   4310   4311   4312   4313
BLKTBL  003352 G      4294#  4931   4969   4994   5037   5087   5124   5401   5643   6476   6557   6577   6682
                      6782   7232
BOE   = 000400 G      3962#
BRKPTR  014566        6244*  6252*  6259*  6310   6324#
BRKTO   014564        6217*  6226*  6232*  6242*  6250*  6314*  6323#
BRKWD   014560        6231   6320#
BUFTBL  027724        6564   7271#
BUF0    031004        7271   7291#
BUF1    032042        7272   7292#
BUF2    033100        7273   7293#
BUF3    034136        7274   7294#
BUF4    035174        7275   7295#
BUF5    036232        7276   7296#
```

```
BUF6    037270          7277    7297#
BUF7    040326          7278    7298#
CARLF   015164          6402    6404    6413#
CHECK   016316          6567    6572#
CHKANR  010722          5641    5653    5657#
CHKANS  010616 G        5062*   5635#
CHKEND  011322 G        5713*   5761*   5798#
CHKERR  011426          5811    5818#
CHKPKS  010726 G        5639*   5651*   5687#
CHKPTR  010724          5643*   5644    5652    5654*   5659#
CHKREE  011500          5804    5825#
CHKRET  012044          5809    5814    5817    5820    5824    5842    5850    5858    5860    5872#
CHKSUC  012340 G        5806*   5825*   5907#
CHKSUM  013674 G        6127#   6170*   6895*   6913*   6944*   6945^   6979*   6980*   7022*   7053*   7084*   7115*   7156*
                        7184*   7211*
CHK8    010636          5638    5643#
CKCKSM  013770 G        5706*   5744*   5756*   6167#
CLRALL  005730 G        4994#   5299*
CLRBUF  005770 G        4996*   5012#   5302*
CLRPTR  006022          4994*   4995    4997    4999*   5022#
CMDSNT= 000100 G        4233#   5222*   5914    5918    5931    6080    6219*
CMNDER= 000040 G        3997#   5952
CMPDAT  002216          3887#   6346
CNINIT= 000032 G        3994#   6272
COMPAR  014570 G        5722*   5746*   6341#
CSNRDY  003346 G        4184#   5267
CSRCVB  003350 G        4185#   5573
C$AU  = 000052          3772#   6826
C$AUTO= 000061          3772#   6693
C$BRK = 000022          3772#   5060    5148    5271    5595    6222    6312    6313    6475    6478    6480
C$BSEG= 000004          3772#
C$BSUB= 000002          3772#
C$CEFG= 000045          3772#
C$CLCK= 000062          3772#
C$CLEA= 000012          3772#   6743
C$CLOS= 000035          3772#
C$CLP1= 000006          3772#
C$CVEC= 000036          3772#   6278    6280    6692
C$DCLN= 000044          3772#   5150    6575    6595
C$DODU= 000051          3772#   6057    6704
C$DRPT= 000024          3772#   6721
C$DU  = 000053          3772#   6781
C$EDIT= 000003          3772#   3815
C$ERDF= 000055          3772#   6011    6025
C$ERHR= 000056          3772#   6038    6053
C$ERRO= 000060          3772#
C$ERSF= 000054          3772#   5147    6574    6594
C$ERSO= 000057          3772#   6031    6049    6362
C$ESCA= 000010          3772#
C$ESEG= 000005          3772#
C$ESUB= 000003          3772#
C$ETST= 000001          3772#   6900    6927    6962    7004    7036    7067    7098    7129    7251
C$EXIT= 000032          3772#   6893    6907    6937    6972    7012    7043    7074    7105    7141
C$GETB= 000026          3772#
C$GETW= 000027          3772#
C$GMAN= 000043          3772#
```

```
C$GPHR= 000042          3772#   6581
C$GPLO= 000030          3772#
C$GPRI= 000040          3772#
C$INIT= 000011          3772#   6662
C$INLP= 000020          3772#
C$MANI= 000050          3772#
C$MEM = 000031          3772#
C$MSG = 000023          3772#   6087
C$OPEN= 000034          3772#
C$PNTB= 000014          3772#   5847    6077    6080    6083    6363
C$PNTF= 000017          3772#   5502    6397    6402    6404    6701    6761    7231    7238    7240    7242
C$PNTS= 000016          3772#   6477    6479    6495    6496    6506
C$PNTX= 000015          3772#   5823    5828    5831    5833    5854    5862
C$QIO = 000377          3772#
C$RDBU= 000007          3772#
C$REFG= 000047          3772#   6553
C$RESE= 000033          3772#   6556    6742
C$REVI= 000003          3772#   3815
C$RFLA= 000021          3772#   6632
C$RPT = 000025          3772#   6519
C$SEFG= 000046          3772#
C$SPRI= 000041          3772#   6237
C$SVEC= 000037          3772#   6238    6240    6681
C$TPRI= 000013          3772#
DESC    014750          6363    6374#
DEVPTR  003314 G        4166#   5087*   5088    5118    5120*   5124*   5125    5141    5143*   6557*   6559    6566    6568*
                        6577*   6579    6626*   7232*   7233    7243    7245*
DEV0    003372          4294    4306#
DEV1    003604          4295    4307#
DEV2    004016          4296    4308#
DEV3    004230          4297    4309#
DEV4    004442          4298    4310#
DEV5    004654          4299    4311#
DEV6    005066          4300    4312#
DEV7    005300          4301    4313#
DFPTBL  002176 G        3860#
DFTL1   013014          6023*   6024*   6025#
DIAGMC= 000000          3772
DLV   = 000074 G        4231#   5358    5444    5475    5576*   5577    5578    5580*   5601*   5602    5603    5605*   5726
                        6081    6083    6084*   6298*   6299    6300    6302*   6623*
DOBRK   014030 G        5111*   5131*   6216#   7164*
DONE    003324 G        4170#   5036*   5056    6896*   6918*   6960*   6995*   7033*   7064*   7095*   7126*   7225*
DR    = 000060 G        4224#   4935    4937    4939*   4971*   4972    4974*   6077    6108    6586*   6590    6592    6913
                        6944    6945    6979    6980    7021    7022    7052    7053    7083    7084    7114    7115
DRVCHK  002220          3888#   7019    7050    7081    7112
DOT3FL  002226          3891#   4950
EF.CON= 000036 G        3962#
EF.NEW= 000035 G        3962#
EF.PWR= 000034 G        3962#
EF.RES= 000037 G        3962#
EF.STA= 000040 G        3962#   6553
ENDGP8  010166          5400    5497    5499#
ENDRSP  007120          5199    5251#
ENDT9   027232          7229    7244    7249#
ERRDES  013210 G        6011    6025    6031    6038    6049    6053    6070#   6362
ERRMOD  010170          5464    5501#
```

```
ESABO  = 177720 G  40540
ESCKS  = 177757 G  40650  4067  4068  4069  5926
ESCKSM= 177757     40670
ESCMD  = 177720 G  40630  5950
ESNCRT= 177767 G  40550  5945
ESNOMO= 177737 G  40620  5921
ESNONX= 177770 G  40560  5955
ESOK   = 000000 G  40570  5909
ESPART= 177776 G  40580  5965
ESRD   = 177757    40690
ESREC  = 177711 G  40640  5970
ESSK   = 177740 G  40590  5939
ESSLF  = 177777 G  40660
ESTRY  = 000001 G  40600  5912
ESWLOC= 177765 G  40610  5960
ESWR   = 177757    40680
EVL    = 000004 G  39620  6019
EVLTHR  002222     38890  6021
EXOFF   007447     53730  5488  5588
EXON    007446     5329   5372#  5417  5596  5608
E$END  = 002100    37720
E$LOAD= 000C55     37720  3815
FLGLOC  016774 G   6019   6632#  6671#
FM      015750     6496   6506   6529#
FMO     015732     6495   6526#
FTLNM   003322     41690  6042
F$AU   = 000015    37720  6803  6826
F$AUTO= 000020     37720  6679  6693
F$BGN  = 000040    37720  3798  3824  3902  3955  6070  6287  6295  6416  6460  6467  6535  6545
                   6679   6716  6753  6803  6888  6891  6893  6900  6905  6907  6927  6935  6937
                   6962   6970  6972  7004  7010  7012  7036  7041  7043  7067  7072  7074  7098
                   7103   7105  7129  7134  7141  7251  7302  7367  7378  7421  7458  7460  7461
                   7466   7467
F$CLEA= 000007     37720  6716  6743
F$DU   = 000016    37720  6753  6781
F$END  = 000041    37720  3798  3902  3955  6087  6291  6304  6416  6460  6519  6535  6662  6693
                   6743   6781  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  7251  7302  7367  7393  7436  7458  7460  7461  7466
                   7467
F$HARD= 000004     37720  7378  7393
F$HW   = 000013    37720  3860  3873
F$INIT= 000006     37720  6545  6662
F$JMP  = 000050    37720  6893  6907  6937  6972  7012  7043  7074  7105  7141
F$MOD  = 000000    37720  3798  3902  3955  6416  6460  6535  6888  7302  7367  7458
F$MSG  = 000011    37720  6070  6087
F$PROT= 000021     37720  3824  3828
F$PWR  = 000017    37720
F$RPT  = 000012    37720  6467  6519
F$SEG  = 000003    37720
F$SOFT= 000005     37720  7421  7436
F$SRV  = 000010    37720  6287  6291  6295  6304
F$SUB  = 000002    37720
F$SW   = 000014    37720  3882  3900
F$TEST= 000001     37720  6891  6900  6905  6927  6935  6962  6970  7004  7010  7036  7041  7066
                   7072   7098  7103  7129  7134  7251
```

```
GBTMP   010612          5572#  5582#  5592#  5593   5612#  5618#
GBTMP2  010614          5587#  5598   5600   5607#  5610   5619#
GETANS  007172 G        5058#  5295#
GETHRD  016340          6573   6577#
GETPTR  007236          5307#
GETRS   017246          6757#  6782#
GTAGIN  007504          5402#  5494
GTBYTE  010366 G        5337#  5357#  5421#  5441#  5472#  5572#
GTDOWN  010124          5406   5410   5425   5436   5440   5443   5447   5474   5478   5480   5486   5490#
GTOK    007750          5429   5449#
GTPKS1  007240 G        5304#  5324#
GTPKS8  007450 G        5300#  5395#  5498
GTPTR   010364          5401#  5402   5491   5493#  5546#
GTUM    007712          5434   5438#
G$CNTO= 000200          3772#
G$DELM= 000372          3772#
G$DISP= 000003          3772#
G$EXCP= 000400          3772#
G$HILT= 000002          3772#
G$LOLI= 000001          3772#
G$NO  = 000000          3772#
G$OFFS= 000400          3772#  7381   7382   7383   7384   7385   7422   7423   7424   7425   7426   7427   7428
                        7429
G$OFSI= 000376          3772#  7381   7382   7383   7384   7385   7422   7423   7424   7425   7426   7427   7428
                        7429
G$PRMA= 000001          3772#  7381   7382
G$PRMD= 000002          3772#  7422   7427
G$PRML= 000000          3772#  7383   7384   7385   7423   7424   7425   7426   7428   7429
G$RADA= 000140          3772#
G$RADB= 000000          3772#
G$RADD= 000040          3772#  7422   7427
G$RADL= 000120          3772#  7383   7384   7385   7423   7424   7425   7426   7428   7429
G$RADO= 000020          3772#  7381   7382
G$XFER= 000004          3772#
G$YES = 000010          3772#  7381   7382   7383   7384   7385   7422   7423   7424   7425   7426   7427   7428
                        7429
HARDR = 000136 G        4261#  6491   6501
HARDW = 000140 G        4262#  6493   6503
HELP  = 000000          3757#  3767   3789   3807   3830   3845   3867   3893   3907#  3945   3964   4314   4320
                        4336   4341   4349   4356   4361   4367   4863   4875   4880   4886   4891   4897   4905
                        4912   4918   4924   6422#  6640   6650   6723   6730   6763   6769   6806   6812   6832#
                        6873   6879   7303   7308   7318   7329#  7387   7403   7430   7451
HOE   = 100000 G        3962#
HRD     013146          6043   6051#
HRDRD = 000016 G        3988#  5865
HRDWR = 000020 G        3989#  5867
HRD1    011766          5852   5861#
IBE   = 010000 G        3962#
IDPTR   003326 G        4171#  5037#  5038   5040   5042#
IDU   = 000040 G        3962#
IER   = 020000 G        3962#
INIT    016204          6547#
INITWD  014562          6216#  6243   6251   6258   6265   6271   6321#
INIT2   016234          6554   6557#
ISR   = 000100 G        3962#
IXE   = 004000 G        3962#
```

```
I$AU  - 000041          3772#  6803#  6826#
I$AUTO- 000041          3772#  6679#  6693#
I$CLN - 000041          3772#  6716#  6743#
I$DU  - 000041          3772#  6753#  6781#
I$HRD - 000041          7378#  7393#
I$INIT- 000041          3772#  6545#  6662#
I$MOD - 000041          3772#  3798#  3902#  3955#  6416#  6460#  6535#  6888#  7302#  7367#  7458#
I$MSG - 000041          3772#  6070#  6087#
I$PROT- 000040          3772#  3824#
I$PTAB- 000041          3772#  7461#  7466#
I$PWR - 000041          3772#
I$RPT - 000041          3772#  6467#  6519#
I$SEG - 000041          3772#  6891   6905   6935   6970   7010   7041   7072   7103   7134
I$SETU- 000041          3772#  7460#  7461   7467#
I$SFT - 000041          7421#  7436#
I$SRV - 000041          3772#  6287#  6291#  6295#  6304#
I$SUB - 000041          3772#  6891   6905   6935   6970   7010   7041   7072   7103   7134
I$TST - 000041          3772#  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   7251#
J$JMP - 000167          3772#
LENGTH  002210          3884#  6634
LGOFST- 000120 G        4251#  6003
LNCNT   015150          6395*  6400*  6409#
LOE   - 040000 G        3962#
LOG     012654 G        5277*  5616*  5709*  5729*  5738*  5766*  5816*  5839*  5868*  5976*  5995#  6229*  6273*
                        6317*
LOGO    013176          6032   6050   6058#
LOGOK   012742          6010   6013#
LOGOK2  013026          6020   6022   6027#
LOGO    012730          6011#
LOG1    013044          6029*  6030*  6031#
LOG2    013074          6036*  6037*  6038#
LOG3    013134          6047*  6048*  6049#
LOG3B   013156          6051*  6052*  6053#
LOT   - 000010 G        3962#
LSTDEV  003370 G        4301#  4941   4975   4997   5040   5118   5141   5491   5652   6507   6566   6688   7243
L$ACP   002110 G        3815#
L$APT   002036 G        3815#
L$AU    017326 G        3815   6803#
L$AUT   002070 G        3815#
L$AUTO  016776 G        3815   6679#
L$CCP   002106 G        3815#
L$CLEA  017160 G        3815   6716#
L$CO    002032 G        3815#
L$DEPO  002011 G        3815#
L$DESC  002122 G        3815   3817#
L$DESP  002076 G        3815#
L$DEVP  002060 G        3815#
L$DISP  002152 G        3815   3843#
L$DLY   002116 G        3815#
L$DTP   002040 G        3815#
L$DTYP  002034 G        3815#
L$DU    017202 G        3815   6753#
L$DUT   002072 G        3815#
L$DVTY  005512 G        3815   4334#
```

```
L$EF    002052 G        3815#
L$ENVI  002014 G        3815#
L$ETP   002102 G        3815#
L$EXP1  002046 G        3815#
L$EXP4  002064 G        3815#
L$EXP5  002066 G        3815#
L$HARD  041366 G        3815    7378#
L$HIME  002120 G        3815#
L$HPCP  002016 G        3815#
L$HPTP  002022 G        3815#
L$HW    002176 G        3815    3860#
L$ICP   002104 G        3815#
L$INIT  016204 G        3815    6545#
L$LADP  002026 G        3815#
L$LAST  042236 G        3815    7457#   7467
L$LOAD  002100 G        3815#
L$LUN   002074 G        3815#   6000*   6001*   6360*   6361*   6580*
L$MREV  002050 G        3815#
L$NAME  002000 G        3815#
L$PRIO  002042 G        3815#
L$PROT  002142 G        3815    3824#
L$PRT   002112 G        3815#
L$REPP  002062 G        3815#
L$REV   002010 G        3815#
L$PPT   015170 G        3815    6467#
L$SOFT  041530 G        3815    7421#
L$SPC   002056 G        3815#
L$SPCP  002020 G        3815#
L$SPTP  002024 G        3815#
L$STA   002030 G        3815#
L$SW    002210 G        3815    3882#
L$TEST  002114 G        3815#
L$TIML  002014 G        3815#
L$UNIT  002012 G        3815#   6572    6628
L10001  002206          3860    3873#
L10002  002230          3882    3900#
L10003  013400          6087#
L10004  014466          6291#
L10005  014522          6304#
L10006  015612          6519#
L10007  016704          6662#
L10010  017102          6693#
L10011  017200          6743#
L10012  017244          6781#
L10013  017326          6826#
L10014  017530          6893    6900#
L10015  020002          6907    6927#
L10016  021374          6937    6962#
L10017  023000          6972    7004#
L10020  023770          7012    7036#
L10021  024554          7043    7067#
L10022  025544          7074    7098#
L10023  026330          7105    7129#
L10024  027234          7141    7251#
L10025  041430          7378    7393#
L10026  041620          7421    7436#
```

```
L10027  042242          7461#
L10031  042252          7461    7466#
MABEE   013106          6035    6041#
MESMRS  010220          5502    5505#
MODRSP  010272          5466    5507#
MRSDLY  010274          5455*   5460*   5509#
MRSP  = 000210 G        4283#   5198    5398    5485    5649    5720*   5760*   6624*   7236
MSAGE1  027236          7231    7253#
MSAGE2  027277          7238    7254#
MSAGE3  027333          7240    7255#
MSAGE4  027362          7242    7256#
MSAUTO  017140          6701    6706#
MSBDA   002342 G        4108#   6362
MSCMD   002706 G        4093    4130#
MSCOM   002406 G        4083    4112#
MSG1    041430          7381    7395#
MSG1B   041441          7382    7396#
MSG1C   041456          7383    7397#
MSG10   042142          7429    7444#
MSG2    041474          7384    7398#
MSG3    041511          7385    7399#
MSG4    041620          7422    7437#
MSG4B   041665          7423    7438#
MSG5    041727          7424    7439#
MSG6    041761          7425    7440#
MSG7    042006          7426    7441#
MSG8    042034          7427    7442#
MSG9    042075          7428    7443#
MSHCHK  002560 C        4086    4122#
MSHDRD  003156 G        4084    4146#
MSHDWR  003220 G        4085    4148#
MSNIT   002622 G        4090    4124#
MSNLOG  002324 G        4077    4098    4106#
MSNOMO  002450 G        4089    4114#
MSNOTP  002466 G        4099    4116#
MSNRSP  002766 G        4097    4136#
MSOVRN  003262 G        4082    4150#
MSPART  002636 G        4091    4126#
MSQRSP  003002 G        4081    4138#
MSREC   002722 G        4094    4132#
MSRNIT  002540 G        4080    4120#
MSSELF  002366 G        4095    4110#
MSSFRD  003056 G        4078    4142#
MSSFWR  003116 G        4079    4144#
MSSKER  002310 G        4087    4104#
MSTOSN  003034 G        4100    4140#
MSUNIT  002660 G        4092    4128#
MSWPRO  002516 G        4088    4118#
MSWRSP  002742 G        4096    4134#
MXRTRY  003332 G        4173#   5851
NCART = 000054 G        4002#   5947
NODRVS  016736          6594    6667#
NOMOR   006554          5147    5154#
NOMOT = 000030 G        3993#   5923
NOREE   011354          5802    5806#
NOUNIT= 000036 G        3996#   5957
```

```
NOXOFF  006652          5200#
NTSFT   013056          6028    6034#
NXTRET  006550          5117    5146    5151#
NXTST   006122 G        5054*   5086#
NXTST2  006372          5119    5123#
ODTFLG= ****** U        6548    7468
ONEFIL= 000001          3309#   3313    3753    3754    3793    3903    3904    3917    6418    6419    6432    6828    6829
                        6840    7324    7325    7339
OTL   = 000010 G        3985#   5725
OVRFLO  013554          6011    6092#
OVRN  = 000012 G        3986#   5728
O$APTS= 000000          3772#   3815
O$AU  = 000001          3772#   3805#   3815
O$BGNR= 000001          3772#   3805#   3815
O$BGNS= 000001          3772#   3805#   3815
O$DU  = 000001          3772#   3805#   3815
O$ERRT= 000000          3772#   3815
O$GNSW= 000001          3772#   3805#   3815
O$POIN= 000001          3772#   3805#   3815
O$SETU= 000001          3772#   3805#   3815    7457
PARTL = 000034 G        3995#   5967
PATTEN= 000072 G        4230#   6079    6353    6619*   6943*   6944    6978*   6979    7018*   7021*   7022    7049*   7052*
                        7080*   7083*   7084    7111*   7114*
PDTFLG  016772 G        6588*   6615    6670#
PERDEV  006404          5125#   5144
PKPTR = 000104 G        4236#   5214*   5420    5479    5481*   5695*   6345    6394
PNT   = 001000 G        3962#
PPSOT9  002224          3890#   7228
PRBUF   002214          3886#   6392
PRDAT   015152          6396*   6397    6410#
PRFORM  015154          6397    6411#
PRI   = 002000 G        3962#
PRI00 = 000000 G        3962#   6237
PRI01 = 000040 G        3962#
PRI02 = 000100 G        3962#
PRI03 = 000140 G        3962#
PRI04 = 000200 G        3962#
PRI05 = 000240 G        3962#
PRI06 = 000300 G        3962#
PRI07 = 000340 G        3815    3962#   6238    6240    6681
PRNPAK  015004 G        6366*   6387#
PRNSIZ  003340 G        4176#   6365*   6398*
PTR     017276          6782*   6783    6786*   6789#
RCBCNT  003320          4168#   5335*   5346*   5351*   5355*   5414*   5433*   5437*   5439*   5470*   5697*   5704    5771
RCBFSZ= 001036 G        4040#   5015    7287    7291    7292    7293    7294    7295    7296    7297    7298
RCDB  = 000024 G        4216#   5576    5601    6236    6298    6611*
RCFLG   003316 G        4167#   5336*   5342    5412*   5428    5698*
RCINIT= 000006 G        3984#   5737
RCSR  = 000022 G        4215#   5463    5574    5590    6260*   6277*   6297*   6609*   6686
RCVBUF= 000102 G        4235#   5014    5214    5332    5689    6564*
RCVHND  014470          6297#
RCVINT  014470 G        6238    6295#
RDNO  = 000114 G        4240#   5232*   6496
RDN1  = 000116 G        4241#   5234*   6506
REC   = 000064 G        4226#   5210    6078    6621*   6911*   6913    6942*   6944    6945    6951*   6952    6955*   6956*
                        6977*   6979    6980    6986*   6987    6990*   6991*   7015*   7018    7022    7025*   7030*   7046*
```

```
                        7049    7053    7056*   7061*   7077*   7080    7084    7087*   7092*   7108*   7111    7115    7118*
                        7123*   7175    7202
RECDAT  017766          6911    6916    6921#
RECERR= 000042 G        3998#   5972
RECID   013462 G        6080    6090#
RECID2  013636          6083    6094#
RECOV   012064          5828    5879#
RETERR  012266          5847    5889#
RETRY = 000002 G        4207#   5822*   5823    5828    5840*   5848*   5851    5853*   5854    5869*   6620*
RLUN    015614          6485*   6486*   6495    6520#
RPTR    015616          6476*   6481    6507    6509*   6521#
RSCMND= 000002 G        4020#   4026    5220    6895    6913    6944    6945    6979    6980    7022    7053    7084    7115
                        7145    7172    7188    7199    7215
RSCONT= 000020 G        4021#   5507    5701    6265    6944    6979    7022    7084
RSDASZ= 000204 G        4033#   4035    4040    5351    5742    5749
RSDATA= 000001 G        4025#   5349    5435    5479    5716    5740    6944    6945    6979    6980    7022    7053    7084
                        7115    7158
RSDNSZ= 000222 G        4035#   5437
RSEND = 000002 G        4026#   5344    5431    5711    5752    6895    6944    6945    6979    6980    7022    7053    7084
                        7115
RSGCDP= 000034 G        4037#   7159
RSINIT= 000004 G        4024#   5734    6321
RSMSIZ= 000012 G        4031#   4039    6895    6913    6944    6945    6979    6980    7022    7053    7084    7115    7146
                        7153    7173    7181    7200    7208
RSNDSZ= 000016 G        4029#   4035    4040    5346    5433    6895    6913    6944    6945    6979    6980    7022    7053
                        7084    7115    7189    7216
RSNTAB  002230          4077#   6016
RSSEND= 000100 G        4044#
RSSGET= 000012 G        4048#   7147
RSSNIT= 000001 G        4050#   6219    7174
RSSNOP= 000000 G        4049#   7201
RSSNSZ= 000016 G        4039#   5754    6895    6913    6944    6945    6979    6980    7022    7053    7084    7115    7155
                        7187    7214
RSSRD = 000002 G        4046#   5227    5914    6944    6945    6979    6980    7022    7053    7084    7115
RSSSEK= 000005 G        4047#   6913
RSSSLF= 000007 G        4051#   5931    6895
RSSWR = 000003 G        4045#   5236    5918    6944    6979    7022    7084
RSVP    006600 G        5191#   6895*   6913*   6944*   6945*   6979*   6980*   7022*   7053*   7084*   7115*   7162*   7192*
                        7219*
RSXOFF= 000023 G        4023#   5373    7284    7285
RSXON = 000020 G        4022#   5372
RTRYN   012224          5823    5854    5885#
RUN     006072 G        5054#   5063    6892*   6906*   6936*   6971*   7011*   7042*   7073*   7104*   7138*
SAVCNT= 000206 G        4282#   5098    5109*
SECREC  003336 G        4175#   6636*   6639*   7030    7061    7092    7123
SERVST  010362          5298*   5495    5531*   5545#
SETDR   005652 G        4969#   6892*   6906*   6936*   6971*   7011*   7042*   7073*   7104*
SETLEN  016646          6634#
SETPTR  005726          4969*   4970    4975    4977*   4980#
SETSRV  010276          5405*   5409*   5521#
SETUP   006024 G        5036#   6892*   6906*   6936*   6971*   7011*   7042*   7073*   7104*   7137*
SFPTBL  002210 G        3882#
SFT     013124          6044    6047#
SFTOUT  041620          7436#
SFTRD = 000002 G        3982#   5836
SFTWR = 000004 G        3983#   5838
```

```
SKERR = 000024 G   3991#  5941
SLFER = 000044 G   3999#  5936
SND     006670     5201   5204#  5208
SNDBYT  007122 G   5204+  5266#  5330+  5419+  5467+  5489+  5589+  5597+  5609+
SNDCNT= 000070 G   4229#  5203+  5207+  6895+  6913+  6944+  6945+  6979+  6980+  7022+  7053+  7084+  7115+
                   7155+  7187+  7214+
SNDHND  014454     6289#
SNDINT  014454 G   6240   6287#
SOFTR = 000122 G   4252#  6487   6497
SOFTW = 000124 G   4253#  6489   6499
SRVTBL  010342     5525   5536#
STAEOP  002212     3885#  6719
STATHD  015620     6477   6523#
STATUS= 000000 G   4206#  7234
STHD2   016074     6479   6532#
STRT    016770 G   6551+  6555+  6597   6669#
SUCCS = 000076 G   4232#  5212+  5810   5826   5908+  6080   6622+
SUCOTL= 000046 G   4000#  5975
SVCGBL= 000000     3772#  3781#  3815   3817   3824   3843   3860   3882   4334   6070   6287   6295   6467
                   6545   6679   6716   6753   6803   7378   7421   7457#
SVCINS= 000001     3772#  3778#  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
SVCSUB= 000001     3772#  3780#
SVCTAG= 000001     3772#  3782#  3873   3900   6087   6291   6304   6519   6662   6693   6743   6781   6826
                   6900   6927   6962   7004   7036   7067   7098   7129   7251   7393   7436   7461   7466
SVCTST= 000001     3772#  3779#  6891   6905   6935   6970   7010   7041   7072   7103   7134
SWAPDR  005532 G   4930#  6892+  6906+  6936+  6971+  7011+  7042+  7073+  7104+
SWPTR   005650     4931+  4932   4941   4943+  4955#
SYSTAT  003310 G   4156#  5100+  5105+  5114+  5296   5341+  5427+  5637   5696+  5801   5803   5872+  6077
                   6129+  6132+  6143   6271+  6631+
S$LSYM= 010000     3772#  3873#  3900#  6087#  6291#  6304#  6519#  6662#  6693#  6743#  6781#  6826#  6900#
                   6927#  6962#  7004#  7036#  7067#  7098#  7129#  7251#  7393#  7436#
TAPLEN  003312 G   4165#  6634+  6635+  6637   7016   7031   7047   7062   7078   7093   7109   7124
TEST9   003344 G   4178#  5196   5396   5415   5452   5483   5647   5718   5758   6552+  7143+  7169+  7226+
THRSHI  012172     5833   5883#
THRSLO  012144     5831   5881#
TMP   = 000066 G   4228#  6940+  6956   6957+  6975+  6991   6992+  7016+  7023+  7031+  7047+  7054+  7062+
                   7078+  7085+  7093+  7109+  7116+  7124+
TOMANY  016706     6574   6665#
TORCVB= 000050 G   4001#  5615   6316
TOSNDB= 000056 G   4003#  5276   6228
TRBUF   027746     5200   5202   5209   6895   6913   6944   6945   6979   6980   7022   7053   7084   7115
                   7144   7171   7198   7287#
TRK   = 000062 G   4225#  7017+  7027   7029+  7048+  7058   7060+  7079+  7089   7091+  7110+  7120   7122+
TRPHND  017106     6681   6701#
TRPPTR  017104     6682+  6683   6688   6690+  6694#
TSTPC = 000020 G   4214#  5039+  5116   5137   5192+
TSTTOP  003330     4172#  4948   5039   6892+  6906+  6936+  6971+  7011+  7042+  7073+  7104+  7136+
```

```
TST1    017374          6892    6895#
TST2    017576          6906    6910#
TST3    020050          4948    6936    6940#
TST3PT  022766          6943    6947    6978    6982    6997#
TST4    021442          6971    6975#
TST5    023046          7011    7015#
TST5EX  023762          7028    7033#
TST6    024036          7042    7046#
TST6EX  024546          7059    7064#
TST7    024622          7073    7077#
TST7EX  025536          7090    7095#
TST8    025612          7104    7108#
TST8EX  026322          7121    7126#
TST9    026354          7136    7143#
TUVECT= 000204 G        4281#   6238    6239*   6240    6241*   6278    6279*   6280    6281*   6285*
T$ARGC= 000002          3815#   5502#   5823#   5828#   5831#   5833#   5847#   5854#   5862#   6077#   6080#   6083#   6363#
                        6397#   6402#   6404#   6477#   6479#   6495#   6496#   6506#   6701#   6761#   7231#   7238#   7240#
                        7242#
T$CODE= 007130          7381#   7382#   7383#   7384#   7385#   7422#   7423#   7424#   7425#   7426#   7427#   7428#   7429#
T$ERRN= 000146          3772#   5147#   6011#   6025#   6031#   6038#   6049#   6053#   6362#   6574#   6594#
T$EXCP= 000000          7381#   7382#   7422#   7427#
T$FLAG= 000040          6893#   6907#   6937#   6972#   7012#   7043#   7074#   7105#   7141#
T$FREE= 042252          7457    7467#
T$GMAN= 000000          3772#
T$HILI= 000376          7381#   7382#   7422#   7427#
T$LAST= 000001          3772#   7457#   7460
T$LOLI= 000001          7381#   7382#   7422#   7427#
T$LSYM= 010000          3772#   3873    3900    6087    6291    6304    6519    6662    6693    6743    6781    6826    6900
                        6927    6962    7004    7036    7067    7098    7129    7251    7393    7436
T$LTNO= 000011          7457#
T$NEST= 177777          3772#   3798#   3824#   3828#   3860#   3873#   3882#   3900#   3902#   3955#   6070#   6087#   6287#
                        6291#   6295#   6304#   6416#   6460#   6467#   6519#   6535#   6545#   6662#   6679#   6693#   6716#
                        6743#   6753#   6781#   6803#   6826#   6888#   6891#   6900#   6905#   6927#   6935#   6962#   6970#
                        7004#   7010#   7036#   7041#   7067#   7072#   7098#   7103#   7129#   7134#   7251#   7302#   7367#
                        7378#   7393#   7421#   7436#   7458
T$NSO = 000000          3798#   3902    3955#   6416    6460#   6535    6545#   6662    6679#   6693    6716#   6743    6753#
                        6781    6803#   6826    6888#   7302    7367#   7458
T$NS1 = 000005          3824#   3828    3860#   3873    3882#   3900    6070#   6087    6287#   6291    6295#   6304    6467#
                        6519    6891#   6900    6905#   6927    6935#   6962    6970#   7004    7010#   7036    7041#   7067
                        7072#   7098    7103#   7129    7134#   7251    7378#   7393    7421#   7436
T$PCNT= 000000          7460#   7461#
T$PTAB= 010030          7461#
T$PTHV= 000001          3815    7467#
T$PTNU= 000001          3772#   7461#   7467
T$SAVL= 177777          3772#
T$SEGL= 177777          3772#
T$SIZE= 000006          7457    7467#
T$SUBN= 000000          3772#   6891#   6905#   6935#   6970#   7010#   7041#   7072#   7103#   7134#
T$TAGL= 177777          3772#
T$TAGN= 010032          3772#   3824#   3860#   3882#   6070#   6287#   6295#   6467#   6545#   6679#   6716#   6753#   6803#
                        6891#   6905#   6935#   6970#   7010#   7041#   7072#   7103#   7134#   7378#   7421#   7460#   7461#
T$TEMP= 000000          3828#   3843#   3873#   3900#   3902#   6087#   6291#   6304#   6416#   6519#   6535#   6662#   6693#
                        6743#   6781#   6826#   6893#   6900#   6907#   6927#   6937#   6962#   6972#   7004#   7012#   7036#
                        7043#   7067#   7074#   7098#   7105#   7129#   7141#   7251#   7302#   7381#   7382#   7383#   7384#
                        7385#   7393#   7422#   7423#   7424#   7425#   7426#   7427#   7428#   7429#   7436#   7458#
T$TEST= 000011          3772#   6891#   6905#   6935#   6970#   7010#   7041#   7072#   7103#   7134#   7457
```

```
T$TSTM= 177777          3772#  5060   5147   5148   5150   5271   5502   5595   5823   5828   5831   5833   5847
                        5854   5862   6011   6025   6031   6038   6049   6053   6057   6077   6080   6083   6087
                        6222   6237   6238   6240   6278   6280   6312   6313   6362   6363   6397   6402   6404
                        6475   6477   6478   6479   6480   6495   6496   6506   6519   6553   6556   6574   6575
                        6581   6594   6595   6632   6662   6681   6692   6693   6701   6704   6721   6742   6743
                        6761   6781   6826   6893   6900   6907   6927   6937   6962   6972   7004   7012   7036
                        7043   7067   7074   7098   7105   7129   7141   7231   7238   7240   7242   7251
T$TSTS= 000001          3772#  6891#  6905#  6935#  6970#  7010#  7041#  7072#  7103#  7134#
T$$AU = 010013          6803#  6826
T$$AUT= 010010          6679#  6693
T$$CLE= 010011          6716#  6743
T$$DAT= 010031          7461#  7466
T$$DU = 010012          6753#  6781
T$$HAR= 010025          7378#  7393
T$$HW = 010001          3860#  3873
T$$INI= 010007          6545#  6662
T$$MSG= 010003          6070#  6087
T$$PC = 000001          7460#  7467
T$$PRO= 010000          3824#
T$$PTA= 010030          7460#  7461#
T$$RPT= 010006          6467#  6519
T$$SOF= 010026          7421#  7436
T$$SRV= 010005          6287#  6291   6295#  6304
T$$SW = 010002          3882#  3900
T$$TES= 010024          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
                        7251
T1      017330 G        3843   6891#
T1TRY = 000146 G        4265#
T2      017532 G        3843   6905#
T3      020004 G        3843   6935#
T4      021376 G        3843   6970#
T4TRY = 000132 G        4259#
T5      023002 G        3843   7010#
T6      023772 G        3843   7041#
T7      024556 G        3843   7072#
T8      025546 G        3843   7103#
T9      026332 G        3843   7134#
UAM   = 000200 G        3962#
UNIT    013402 G        6077   6088#
UNITNO  027412          5502   7230*  7238   7240   7242   7246*  7258#
UNREC   012244          5862   5887#
UNSUC   011642          5827   5844#
UNXPCT  007654          5430#
WAIT    014524          6246*  6254*  6261*  6308#  6315
WHCHDR  013660 G        5230*  5244*  6006*  6106#
WRLOCK= 000026 G        3992#  5962   6034
WRTN0 = 000110 G        4238#  5246*  6496   6601   6603
WRTN1 = 000112 G        4239#  5249*  6506
XFNSND  006660          5197   5202#
XMDB  = 000030 G        4218#  5279*  6231*  6290*  6618*
XMSR  = 000026 G        4217#  5268   6218*  6224   6235*  6245*  6253*  6276*  6289*  6613*
XSCNT = 000036 G        4221#  6895*  6913*  6944*  6979*  7022*  7084*  7159*  7189*  7216*
XSFLG = 000034 G        4220#  5215   5325   5691   6895*  6913*  6944*  6945   6979*  6980   7022*  7053   7084*
                        7115   7158*  7188*  7215*
xSPKNM= 000032 G        4219#  5211*  5333   5407   5411*  5424*  5430*  5446*  5477*  5690   6895*  6913*  6944*
```

```
                        5945#   6979#   6980#   7022#   7053#   7084#   7115#   7160#   7190#   7217#
XSPTR = 000106 G        4237@   5217#   5412    5413#   5414    5490#
X$ALWA= 000000          3772@
X$FALS= 000040          3772@
X$OFFS= 000400          3772@
X$TRUE= 000020          3772@
.     = 042252          3795@   3817@   4105@   4107@   4109@   4117@   4119@   4121@   4123@   4125@   4129@   4133@   4135@
                        4139@   4141@   4143@   4145@   4147@   4149@   4151@   4306@   4307@   4308@   4309@   4310@   4311@
                        4312@   4313@   5155@   5508@   5884@   6091@   6093@   6412@   6414@   6525@   6527@   6531@   6534@
                        6666@   6668@   6792@   6893    6907    6937    6972    7012    7043    7074    7105    7141    7257@
                        7287@   7291@   7292@   7293@   7294@   7295@   7296@   7297@   7298@   7461    7467
```

BCOMPL	105#	3772#									
BERROR	113#	3772#									
BGNAU	121#	3772#	6803								
BGNAUT	135#	3772#	6679								
BGNCLN	149#	3772#	6716								
BGNDU	162#	3772#	6753								
BGNHRD	175#	3772#	7378								
BGNHW	190#	3772#	3860								
BGNINI	206#	3772#	6545								
BGNMOD	218#	3772#	3798	3955	6460	6888	7367				
BGNMSG	235#	3772#	6070								
BGNPRO	248#	3772#	3824								
BGNPTA	262#	3772#	7461								
BGNRPT	297#	3772#	6467								
BGNSEG	311#	3772#									
BGNSET	325#	3772#	7460								
BGNSFT	352#	3772#	7421								
BGNSRV	367#	3772#	6287	6295							
BGNSUB	380#	3772#									
BGNSW	408#	3772#	3882								
BGNTST	423#	3772#	6891	6905	6935	6970	7010	7041	7072	7103	7134
BNCOMP	457#	3772#	6554	6582							
BNERRO	465#	3772#									
BREAK	473#	3772#	5060	5148	5271	5595	6222	6312	6313	6475	6478 6480
BRESET	481#	3772#	6556	6742							
CKLOOP	491#	3772#									
CLOCK	503#	3772#									
CLOSE	514#	3772#									
CLRVEC	528#	3772#	6278	6280	6692						
COMMEN	537#	3772#									
DELAY	568#	3772#									
DESCRI	558#	3772#	3817								
DEVTYP	592#	3772#	4334								
DISPAT	601#	3772#	3843								
DISPLA	619#	3772#									
DOCLN	639#	3772#	5150	6575	6595						
DODU	647#	3772#	6057	6704							
DORPT	655#	3772#	6721								
ENDAU	663#	3772#	6826								
ENDAUT	679#	3772#	6693								
ENDCLN	696#	3772#	6743								
ENDCOM	712#	3772#									
ENDDU	734#	3772#	6781								
ENDHRD	753#	3772#	7393								
ENDHW	769#	3772#	3873								
ENDINI	783#	3772#	6662								
ENDMOD	800#	3772#	3902	6416	6535	7302	7458				
ENDMSG	817#	3772#	6087								
ENDPRO	834#	3772#	3828								
ENDPTA	848#	3772#	7466								
ENDRPT	861#	3772#	6519								
ENDSEG	880#	3772#									
ENDSET	899#	3772#	7467								
ENDSFT	919#	3772#	7436								
ENDSRV	936#	3772#	6291	6304							
ENDSUB	956#	3772#									

```
ENDSW    978#  3772#  3900
ENDTST   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$CNTO  3291#  3772#  7381#  7382#  7383#  7384#  7385#  7422#  7423#  7424#  7425#  7426#  7427#  7428#  7429#
M$COUN  3136#  3772#  5502#  5823#  5828#  5831#  5833#  5847#  5854#  5862#  6077#  6080#  6083#  6363#  6397#
        6402#  6404#  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#  6826#  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#  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$ESCS  2943#  3772#
M$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#  6304#  6467#  6519#  6545#  6662#  6679#  6693#  6716#  6743#  6753#  6781#  6803#  6826#  6891#
        6900#  6905#  6927#  6935#  6962#  6970#  7004#  7010#  7036#  7041#  7067#  7072#  7098#  7103#  7129#
        7134#  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#  6826#  6900#  6927#  6962#  7004#  7036#  7067#  7098#  7129#  7251#  7302#  7393#  7436#  7458#
```

M$GETT 26450 37720 68930 69070 69370 69720 70120 70430 70740 71050 71410
M$GNGB 27000 37720 37980 38150 38170 38240 38430 38600 38820 39550 43340 60700 62870 62950 64600
64670 65450 66790 67160 67530 68030 68880 73670 73780 74210 74570
M$GNIN 31130 37720 38150 38170 38430 38600 38820 43340 50600 51470 51480 51500 52710 55020 55950
58230 58280 58310 58330 58470 58540 58620 60110 60250 60310 60380 60490 60530 60570 60770
60800 60830 60870 62220 62370 62380 62400 62780 62800 62910 63040 63120 63130 63620 63630
63970 64020 64040 64750 64770 64780 64790 64800 64950 64960 65060 65190 65530 65540 65560
65740 65750 65810 65820 65940 65950 66320 66620 66810 66920 66930 67010 67040 67210 67420
67430 67610 68260 68930 69000 69070 69270 69370 69620 69720 70040 70120 70360 70430
70670 70740 70980 71050 71290 71410 72310 72380 72400 72420 72510 73780 73810 73820 73830
73840 73850 73930 74210 74220 74230 74240 74250 74260 74270 74280 74290 74360 74570 74610
M$GNLS 27280 37720
M$GNSU 26900 37720
M$GNTA 26700 37720 38730 39000 60870 62910 63040 65190 66620 66930 67430 67810 68260 69000 69270
69620 70040 70360 70670 70980 71290 72510 73930 74360 74610 74660
M$GNTE 26800 37720 68910 69050 69350 69700 70100 70410 70720 71030 71340
M$HAPT 24840 37720 38150
M$HNAP 25770 37720 38150
M$INCR 30660 37720 37980 38240 38600 38820 39550 50600 51470 51480 51500 52710 55020 55950 58230
58280 58310 58330 58470 58540 58620 60110 60250 60310 60380 60490 60530 60570 60700 60770
60800 60830 60870 62220 62370 62380 62400 62780 62800 62870 62950 63120 63130 63620 63630
63970 64020 64040 64600 64670 64750 64770 64780 64790 64800 64950 64960 65060 65190 65450
65530 65560 65740 65750 65810 65940 65950 66320 66620 66790 66810 66920 66930 67010 67040
67160 67210 67420 67430 67530 67610 67810 68030 68260 68880 68910 68930 69000 69050 69070
69270 69350 69370 69620 69700 69720 70040 70100 70120 70360 70410 70430 70670 70720 70740
70980 71030 71050 71290 71340 71410 72310 72380 72400 72420 72510 73670 73780 74210 74600
74610
M$IOSE 24380 37720
M$LDRO 27820 37720 60570 62370 62780 62800 65530 65810 66920 67040
M$MASK 23970 37720
M$MCHI 900 37720
M$MCLO 23340 37720
M$MSK1 24090 37720
M$POP 26570 37720 38280 38730 39000 39020 60870 62910 63040 64160 65190 65350 66620 66930 67430
67810 68260 69000 69270 69620 70040 70360 70670 70980 71290 72510 73020 73930 74360 74580
M$PRIN 23560 37720 55020 58230 58280 58310 58330 58470 58540 58620 60770 60800 60830 63630 63970
64020 64040 64770 64790 64950 64960 65060 67010 67610 72310 72380 72400 72420
M$PUSH 23440 37720 37980 38240 38600 38820 39550 60700 62870 62950 64600 64670 65450 66790 67160
67530 68030 68880 68910 69050 69350 69700 70100 70410 70720 71030 71340 73670 73780 74210
M$PUT 28300 37720 55020 58230 58280 58310 58330 58470 58540 58620 60770 60800 60830 62380 62400
63630 63970 64020 64040 64770 64790 64950 64960 65060 66810 67010 67610 72310 72380 72400
72420
M$PUT1 28530 37720 55020 58230 58280 58310 58330 58470 58540 58620 60770 60800 60830 62380 62400
63630 63970 64020 64040 64770 64790 64950 64960 65060 66810 67010 67610 72310 72380 72400
72420
M$RADI 31630 37720 73810 73820 73830 73840 73850 74220 74230 74240 74250 74260 74270 74280 74290
M$RBRO 27980 37720
M$RNRO 28130 37720 65810 66320
M$SETS 30830 37720 37980 38240 38600 38820 39550 60700 62870 62950 64600 64670 65450 66790 67160
67530 68030 68880 68910 69050 69350 69700 70100 70410 70720 71030 71340 73670 73780 74210
M$STAR 24750 37720
M$SVC 27570 37720 50600 5147 51480 51500 52710 55020 55950 58230 58280 58310 58330 58470 58540
58620 6011 6025 6031 6038 6049 6053 60570 60770 60800 60830 60870 62220 62370 62380
62400 62780 62800 63120 63130 6362 63630 63970 64020 64040 64750 64770 64780 64790 64800
64950 64960 65060 65190 65530 65560 6574 65750 65810 6594 65950 66320 66620 66810 66920
66930 67010 67040 67210 67420 67430 67610 67810 68260 68930 69000 69070 69270 69370 69620

```
        6972#  7004#  7012#  7036#  7043#  7067#  7074#  7098#  7105#  7129#  7141#  7231#  7238#  7240#  7242#
        7251#
M$TLAB  2750#  3772#  5060#  5147#  5148#  5150#  5271#  5502#  5595#  5823#  5828#  5831#  5833#  5847#  5854#
        5862#  6011#  6025#  6031#  6038#  6049#  6053#  6057#  6077#  6080#  6083#  6087#  6222#  6237#  6238#
        6240#  6278#  6280#  6312#  6313#  6362#  6363#  6397#  6402#  6404#  6475#  6477#  6478#  6479#  6480#
        6495#  6496#  6506#  6519#  6553#  6556#  6574#  6575#  6581#  6594#  6595#  6632#  6662#  6681#  6692#
        6693#  6701#  6704#  6721#  6742#  6743#  6761#  6781#  6826#  6893#  6900#  6907#  6927#  6937#  6962#
        6972#  7004#  7012#  7036#  7043#  7067#  7074#  7098#  7105#  7129#  7141#  7231#  7238#  7240#  7242#
        7251#
M$TSTL  2739#  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#  6574#  6575#  6581#  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#
M$WORD  2899#  3772#  3815#  3843#  5147#  6011#  6025#  6031#  6038#  6049#  6053#  6362#  6574#  6594#  6893#
        6907#  6937#  6972#  7012#  7043#  7074#  7105#  7141#  7381#  7382#  7383#  7384#  7385#  7422#  7423#
        7424#  7425#  7426#  7427#  7428#  7429#  7461#
M$XFER  2417#  3772#
OPEN    1699#  3772#
POINTE  1707#  3772#  3805
POP     4390#  5019   5020   5272   5280   5468   5532   5533   5874   5875   6058   6059   6060   6061   6085
        6086   6151   6152   6189   6190   6367   6368   6369   6405   6406   6513   6514   6515   6516   6517
        6518   6759   6760
PRINTB  1773#  3772#  5847   6077   6080   6083   6363
PRINTF  1816#  3772#  5502   6397   6402   6404   6701   6761   7231   7238   7240   7242
PRINTS  1859#  3772#  6477   6479   6495   6496   6506
PRINTX  1902#  3772#  5823   5828   5831   5833   5854   5862
PUSH    4377#  5012   5013   5266   5270   5454   5521   5522   5799   5800   5995   5996   5997   5998   6071
        6072   6127   6128   6167   6168   6341   6342   6343   6390   6391   6468   6469   6470   6471   6472
        6473   6755   6756
READBU  1945#  3772#
READEF  1954#  3772#  6553
RFLAGS  1972#  3772#  6632
SETPRI  1982#  3772#  6237
SETVEC  1991#  3772#  6238   6240   6681
SLASH   2004#  3772#
STARS   2021#  3772#
SVC     2042#  3771#  3772
SWAPIN  4407#  5097   5108   5115   5136
SWAPOW  4428#  5193
TSTID   4832#  6892   6906   6936   6971   7011   7042   7073   7104
TUREAD  4714#  6945   6980   7053   7115
TURTRY  4639#  6944   6945   6979   6980   7022   7053   7084   7115
TUSEEK  4577#  6913
TUSELF  4791#  6895
TUWRIT  4469#  6944   6979   7022   7084
XFER    2306#  3772#  6893#  6907#  6937#  6972#  7012#  7043#  7074#  7105#  7141#
XFERF   2317#  3772#
XFERT   2326#  3772#

. ABS.  042252   000
```

```
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

 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)
```

SEQ 0146