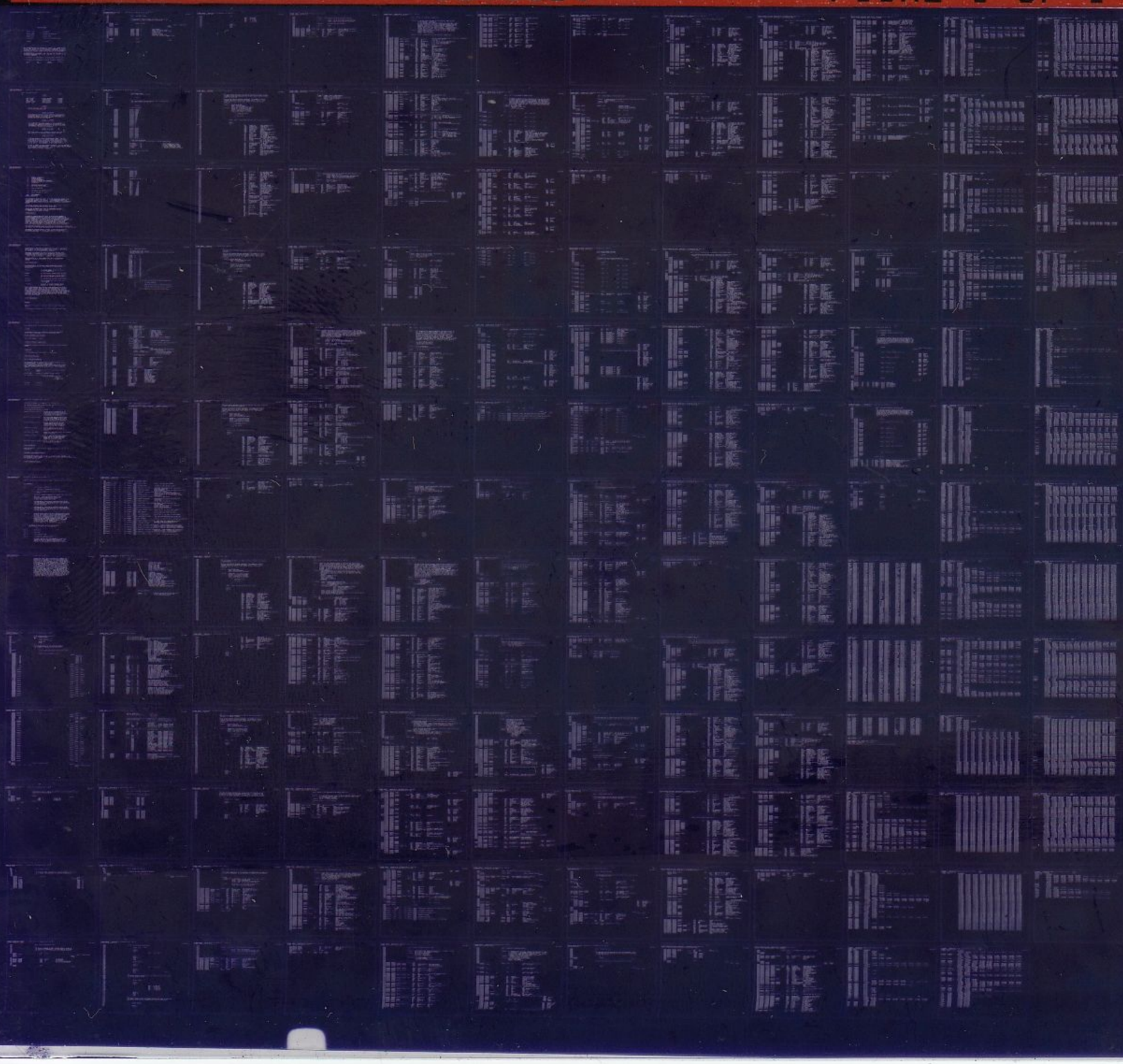
TU58 PERF EXERCISER
CZTUUDO

AH-E649D-MC
FICHE 1 OF 1



.REM &

IDENTIFICATION

PRODUCT CODE: AC-E648D-MC
PRODUCT NAME: CZTUUDO TU58 PERF EXER
PRODUCT DATE: FEBRUARY 1982
MAINTAINER: DIAGNOSTIC ENGINEERING GROUP

AUTHOR: R. J. ROSS

&

```
.SBTTL  REVISION HISTORY
```

.REM &

HISTORY

JUNE 18,1979	INITIAL RELEASE	CZTU A0
JULY 1,1979	SECOND RELEASE	CZTUUB0
JUNE 1,1980	THIRD RELEASE	CZTUUB1
OCTOBER 1,1981	FOURTH RELEASE	CZTUUC0
MARCH 1,1982	FIFTH RELEASE	CZTUUD0

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-E650B-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.

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 SCHEDULING 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 DR0 - 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 RETRIEVED WITH THE "PRI" COMMAND.

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

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 PACKE~ 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. BYTES 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          002000          .       =       2000
                                .NLIST  BEX
002000                          BGNMOD

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

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

                                        HEADER  CZTUU,D,0,3600.,1,PRI07
002000                                                                  L$NAME::
002000
002000      103                                                                 .ASCII  /C/
002001      132                                                                 .ASCII  /Z/
002002      124                                                                 .ASCII  /T/
002003      125                                                                 .ASCII  /U/
002004      125                                                                 .ASCII  /U/
002005      000                                                                 .BYTE   0
002006      000                                                                 .BYTE   0
002007      000                                                                 .BYTE   0
002010                                                                  L$REV::
002010      104                                                                 .ASCII  /D/
002011                                                                  L$DEPO::
002011      060                                                                 .ASCII  /0/
002012                                                                  L$UNIT::
002012  000001                                                                  .WORD   T$PTHV
002014                                                                  L$TIML::
002014  007020                                                                  .WORD   3600.
002016                                                                  L$HPCP::
002016  041342                                                                  .WORD   L$HARD
002020                                                                  L$SPCP::
002020  041504                                                                  .WORD   L$SOFT
002022                                                                  L$HPTP::
002022  002176                                                                  .WORD   L$HW
002024                                                                  L$SPTP::
002024  002210                                                                  .WORD   L$SW
002026                                                                  L$LADP::
002026  042150                                                                  .WORD   L$LAST
002030                                                                  L$STA::
002030  000000                                                                  .WORD   0
002032                                                                  L$CO::
002032  000000                                                                  .WORD   0
002034                                                                  L$DTYP::
002034  000001                                                                  .WORD   1
002036                                                                  L$APT::
002036  000000                                                                  .WORD   0
002040                                                                  L$DTP::
002040  002152                                                                  .WORD   L$DISPATCH
002042                                                                  L$PRIO::
002042  000340                                                                  .WORD   PRI07
002044                                                                  L$ENVI::
```

```
002044  000000
002046
002046  000000
002050
002050     003
002051     003
00205
002052  000000
002054  000000
002056
002056  000000
002060
002060  005510
002062
002062  015150
002064
002064  000000
002066
002066  000000
002070
002070  017302
002072
002072  017156
002074
002074  000000
002076
002076  002122
002100
002100  104035
002102
002102  000000
002104
002104  016164
002106
002106  017136
002110
002110  016754
002112
002112  002142
002114
002114  000000
002116
002116  000000
002120
002120  000000
508
509 002122                      DESCRIP <TU58 PERF EXER>
    002122
    002122     124     125     065
```

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

L$DESC::
                .ASCIZ  /TU58 PERF EXER/
                .EVEN
```

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

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

```
                                .SBTTL  DISPATCH TABLE

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

002150                                  DISPATCH 9
002150  000011                                                          .WORD   9
002152                                                          L$DISPATCH::
002152  017304                                                          .WORD   T1
002154  017506                                                          .WORD   T2
002156  017760                                                          .WORD   T3
002160  021352                                                          .WORD   T4
002162  022756                                                          .WORD   T5
002164  023746                                                          .WORD   T6
002166  024532                                                          .WORD   T7
002170  025522                                                          .WORD   T8
002172  026306                                                          .WORD   T9
```

```
                                .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
002174  000004                                                                 .WORD   L10001-L$HW/2
002176                                                                  L$HW::
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
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
002206  000007                                                                 .WORD   L10002-L$SW/2
002210                                                                  L$SW::
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                                  ENDSW
002226                                                          L10002:

002226                                  ENDMOD
```

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


002226                                  BGNMOD

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

002226                                  EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
                100000          BIT15== 100000
                040000          BIT14== 40000
                020000          BIT13== 20000
                010000          BIT12== 10000
                004000          BIT11== 4000
                002000          BIT10== 2000
                001000          BIT09== 1000
                000400          BIT08== 400
                000200          BIT07== 200
                000100          BIT06== 100
                000040          BIT05== 40
                000020          BIT04== 20
                000010          BIT03== 10
                000004          BIT02== 4
                000002          BIT01== 2
                000001          BIT00== 1
                                ;
                001000          BIT9==  BIT09
                000400          BIT8==  BIT08
                000200          BIT7==  BIT07
                000100          BIT6==  BIT06
                000040          BIT5==  BIT05
                000020          BIT4==  BIT04
                000010          BIT3==  BIT03
                000004          BIT2==  BIT02
                000002          BIT1==  BIT01
                000001          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
                000040          EF.START==      32.                     ; START COMMAND WAS ISSUED
                000037          EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
                000036          EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
                000035          EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
                000034          EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
                                ;
                                ;
                                ; PRIORITY LEVEL DEFINITIONS
                                ;
                000340          PRI07== 340
                000300          PRI06== 300
```

GLOBAL EQUATES SECTION

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

654

```
                        .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
                        ;-----------------------------------------------------------------------
```

```
                                .SBTTL  ERROR MESSAGE DESCRIPTIONS

                                ;THE TABLE OF ERROR MESSAGES (ADDRESSES).  ABNDX(R5) CONTAINS THE OFFSET
                                ;OF THE REASON.  IT'S ABSOLUTE ADDRESS IS RSNTAB + ABNDX(R5).

002226  002322                  RSNTAB: MSNLOG
002230  003054                          MSSFRD
002232  003114                          MSSFWR
002234  002536                          MSRNIT
002236  003000                          MSQRSP
002240  003260                          MSOVRN
002242  002404                          MSCOM
002244  003154                          MSHDRD
002246  003216                          MSHDWR
002250  002556                          MSHCHK
002252  002306                          MSSKER
002254  002514                          MSWPRO
002256  002446                          MSNOMO
002260  002620                          MSNIT
002262  002634                          MSPART
002264  002656                          MSUNIT
002266  002704                          MSCMD
002270  002720                          MSREC
002272  002364                          MSSELF
002274  002740                          MSWRSP
002276  002764                          MSNRSP
002300  002322                          MSNLOG
002302  002464                          MSNOTP
002304  003032                          MSTOSN
```

```
                                        ;HERE ARE THE MESSAGES PROPER:

002306  123  105  105  MSSKER:: .ASCIZ /SEEK ERROR/              ;DEVICE COULD NOT READ HEADER
                              .EVEN
002322  123  131  123  MSNLOG:: .ASCIZ /SYSTEM ERROR/            ;DIAGNOSTIC HUNG. BETTER RE-BOOT
                              .EVEN
002340  102  101  104  MSBDA:: .ASCIZ  /BAD DATA IN PACKET/      ;HOST DATA CHECK FOUND ERROR, DEVICE MAY
                              .EVEN                               ;HAVE READ CORRECTLY.
002364  123  105  114  MSSELF:: .ASCIZ /SELF TEST ERROR/         ;MICRO DIAGNOSTIC FAILED, BUT DEVICE COULD STILL
                              .EVEN                               ;SEND AN END PACKET.
002404  102  101  104  MSCOM:: .ASCIZ  /BAD DATA W-O DATA CHECK ERR AT TU/  ;PREVIOUS DATA CHECK
                              .EVEN                               ;ERROR NOT DUE TO DEVICE READ OPERATION
002446  115  117  124  MSNOMO:: .ASCIZ /MOTOR STOPPED/           ;DEVICE COULD NOT GET ANY MEANINGFUL SIGNAL
                              .EVEN                               ;FROM TAPE.
002464  103  101  122  MSNCTP:: .ASCIZ /CARTRIDGE NOT IN PLACE/  ;NO MEDIA OR BAD SWITCH
                              .EVEN
002514  127  122  111  MSWPRO:: .ASCIZ /WRITE PROTECTION/        ;CARTRIDGE WRITE PROTECT TAB MISSING OR
                              .EVEN                               ;SWITCH BAD
002536  122  105  103  MSRNIT:: .ASCIZ /RECIEVING INIT/          ;DEVICE SENT INIT REQUEST
                              .EVEN
002556  110  117  123  MSHCHK:: .ASCIZ /HOST FOUND PACKET CHECKSUM ERROR/  ;DEVICE SENT PACK WITH
                              .EVEN                               ;BAD CHECKSUM
002620  103  101  116  MSNIT:: .ASCIZ  /CAN'T INIT/              ;DEVICE SENT BYTE OTHER THAN ''CONTINUE''
                              .EVEN                               ;DURING INITIALIZATION
002634  120  101  122  MSPART:: .ASCIZ /PARTIAL OPERATION/       ;END OF MEDIUM ENCOUNTERED
                              .EVEN
002656  042  116  117  MSUNIT:: .ASCIZ /''NON-EXISTENT'' DRIVE/  ;DEVICE RECV'D TOO LARGE DRIVE NUMBER
                              .EVEN
002704  102  101  104  MSCMD:: .ASCIZ  /BAD COMMAND/             ;DEVICE COULD NOT UNDERSTAND HOST
                              .EVEN
002720  102  101  104  MSREC:: .ASCIZ  /BAD RECORD NO./          ;DEVICE RECV'D TOO LARGE A RECORD NUMBER
                              .EVEN
002740  127  122  117  MSWRSP:: .ASCIZ /WRONG SUCCESS CODE/      ;HOST COULD NOT DECIPHER CODE IN END PACK
                              .EVEN
002764  116  117  040  MSNRSP:: .ASCIZ /NO RESPONSE/            ;TIME OUT WAITING FOR BYTE IN RCV BUF ON INTERFACE.
                              .EVEN
003000  111  116  104  MSQRSP:: .ASCIZ \INDECIPHERABLE FLAG BYTE\  ;HOST COULD NOT UNDERSTAND 1ST BYTE OF
                              .EVEN                               ;RESPONSE FROM TU AS PROPER PROTOCOL
003032  124  111  115  MSTOSN:: .ASCIZ /TIME OUT ON SEND/        ;DLV 'READY' NEVER WENT HIGH
                              .EVEN
003054  122  105  103  MSSFRD:: .ASCIZ /RECOV. DATA CHECK ERR ON RD OP/  ;TU58 RESPONDED WITH 'DATA-CHECK''
                              .EVEN                               ;ERROR ON READ OP. ;HOST RETRY(S) SUCCESSFUL
003114  122  105  103  MSSFWR:: .ASCIZ /RECOV. DATA CHECK ERR ON WR OP/  ;SAME BUT WR OR WR VERIFY OPERATION
                              .EVEN
003154  125  116  122  MSHDRD:: .ASCIZ /UNRECOV. DATA CHECK ERR ON RD OP/  ;TU58 RESPONDED WITH 'DATA-CHECK''
                              .EVEN                               ;ERROR ON READ OP.  ;RETRIES UNSUCCESSFUL
003216  125  116  122  MSHDWR:: .ASCIZ /UNRECOV. DATA CHECK ERR ON WR OP/  ;SAME BUT WR OPERATION
                              .EVEN
003260  104  114  126  MSOVRN:: .ASCIZ /DLV ERROR IN RECEIVE/  ;DLV ERROR (THE CONTENTS PRINTED OUT)
                              .EVEN
```

```
                                .SBTTL  MISC STORAGE AND ECUATES

003306  000000          SYSTAT:: .WORD                  ;SYSTEM STATUS WORD
                                                        ;BIT7-BIT15 = 1ST BYTE OF PACK RCVD
                                                        ;BIT05 = NOT USED
                                                        ;BIT04 = NOT USED
                                                        ;BIT03 = NOT USED
                                                        ;BIT02 =RETRY BAD FLAG BYTE
                                                        ;BIT01 = RETRY OCCURRING
                                                        ;BIT00 = CHKSUM IS ODD (TEMP STORAGE)

003310  000000          TAPLEN:: .WORD                  ;# RECORDS
003312  000000          DEVPTR:: .WORD                  ;->NEXT UNIT ADDRESS
003314  000000          RCFLG::  .WORD                  ;TEMP STORE FOR GTBYTE
003316  000000          RCBCNT:  .WORD                  ;TEMP STORAGE FOR GTBYTE
003320  000004          FTLNM:   .WORD 4                ;NUMBER OF TRIES BEFORE ABORT
003322  000000          DONE::   .WORD                  ;1 IF TEST ALGORITHM COMPLETE
003324  000000          IDPTR::  .WORD                  ;PTR IN TESTID MACRO
003326  000000          TSTTOP:  .WORD                  ;POINTER TO CURRENT TEST
003330  000010          MXRTRY:: .WORD  8.              ;NUMBER OF RETRIES, BEFORE "HARD" ERROR
003332  000000          BLKER::  .WORD                  ;RECORD # FOR ERROR REPORT
003334  000000          SECREC:: .WORD                  ;RECORD # TO START AFTER SWITCHING TRACKS
003336  000000          PRNSIZ:: .WORD                  ;SIZE OF PACK FOR "PRNPAK"
003340  000000          ALLGON:: .WORD                  ;SECONDARY DON'T PRINT STATISTICS FLAG.
003342  000000          TEST9::  .WORD                  ;***** FLAG THAT INDICATES IF IN TST 9


                        ;               TIME OUT CONSTANTS:

003344  177777          CSNRDY::  -1            ;TIME OUT ON SEND DELAY (DLV NOT READY)
003346  000226          CSRCVB::  150.          ;45 SEC. MULTIPLIER  REWIND TIME
                                                ;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==VERIFY OP.

000102                  RCVBUF  ==      66.             ;POINTER TO 542. BYTE BUFFER (4 DATA PAKS + END PACK)
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 OPERATION
000210                  MRSP    -=      136.            ;***** FLAG INDICATING MRSP
000212                  BLKSIZ  ==      138.            ;** RESERVED **
                        ;-------------------------------------------------------------------
```

```
                                .SBTTL  DEVICE DATA BLOCK ALLOCATION


                                ;TABLE OF DEVICE DATA BLOCK ADDRESSES

003350  003370                BLKTBL::        .WORD   DEV0
003352  003602                                .WORD   DEV1
003354  004014                                .WORD   DEV2
003356  004226                                .WORD   DEV3
003360  004440                                .WORD   DEV4
003362  004652                                .WORD   DEV5
003364  005064                                .WORD   DEV6
003366  005276                LSTDEV::        .WORD   DEV7


                                ;AND STORAGE FOR EACH:

003370                          DEV0:           .BLKB   BLKSIZ
003602                          DEV1:           .BLKB   BLKSIZ
004014                          DEV2:           .BLKB   BLKSIZ
004226                          DEV3:           .BLKB   BLKSIZ
004440                          DEV4:           .BLKB   BLKSIZ
004652                          DEV5:           .BLKB   BLKSIZ
005064                          DEV6:           .BLKB   BLKSIZ
005276                          DEV7:           .BLKB   BLKSIZ
```

```
                                .SBTTL  GLOBAL TEXT SECTION

                                ;
                                ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;
                                        DEVTYP  <TU58 CONTROLLER>

005510                                                          L$DVTYP::
005510                                                                  .ASCIZ  /TU58 CONTROLLER/
005510  124     125     065                                             .EVEN
```

```
                        .SBTTL  SYSTEM MACRO DEFINITIONS

                                .MACRO  PUSH    ,REG

                                .NLIST
                                .LIST ME
                                .LIST
                                                MOV     REG,-(SP)


                                .NLIST
                                .NLIST ME
                                .LIST
                                .ENDM

                                .MACRO  POP,REG

                                .NLIST
                                .LIST ME
                                .LIST
                                                MOV     (SP)+,REG

                                .NLIST
                                .NLIST ME
                                .LIST
                                .ENDM

                        ;++
                        ;THE MACRO 'SWAPIN' RETRIEVES THE TEST REGISTERS WHICH WERE SAVED
                        ;IN THE DEVICE DATA BLOCK.
                        ;--

                                .MACRO  SWAPIN

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

                                .NLIST
                                .NLIST ME
                                .LIST
                                .ENDM

                        ;++
                        ;THE MACRO 'SWAPOW' SAVES THE CURRENT STATE OF THE UNIT IN THE DRIVE
                        ;DATA BLOCK IN SO THAT THE SCHEDULER MAY 'SWAPIN' ANOTHER UNIT.
                        ;--

                                .MACRO  SWAPOW

                                .NLIST
                                .LIST ME
```

```
1125                                    .LIST
1126                                                    MOV     R0,6.(R5)
1127                                                    MOV     R1,8.(R5)
1128                                                    MOV     R2,10.(R5)
1129                                                    MOV     ,12.(R5)
1130                                                    MOV     R4,14.(R5)
1131
1132                                    .NLIST
1133                                    .NLIST ME
1134                                    .LIST
1135                                    .ENDM
```

```
                        ;++
                        ;THE WRITE MACRO IMPLEMENTS THE COMPLETE PROTOCOL NECESSARY TO BUILD
                        ;A COMMAND PACKET AND SUBSEQUENT DATA PACKETS (UNTIL THE BYTE COUNT
                        ;(BCNT) IS SATISFIED).
                        ;
                        ;SETS UP THE EXPECTED PROTOCOL RESPONSES:  THE NUMBER OF PACKETS
                        ;(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
                        ;           UNIT'S 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  TUWRIT  PTRN,REC,BCNT,DR,VER,?A,?B,?C,?D,?E,?F,?G,?H,?T

                                .NLIST
                                .LIST ME
                                .LIST
                        T:      MOV     #TRBUF,R0       ;MAKE COMMAND PACKET:
                                MOVB    #RSCMND,@R0     ;COMMAND FLAG
                                MOVB    #RSMSIZ,1(R0)   ;THIS SIZE
                                MOVB    #RSSWR,2(R0)    ;INSERT OP CODE-WRITE
                                MOVB    VER,3.(R0)      ;VERIFY (1 OR 0)
                                MOVB    DR,4.(R0)       ;DRIVE #
                                MOVB    #020,5.(R0)     ;MAINTENANCE MODE SWITCH
                                CLR     6.(R0)          ;NO SEQUENCE #
                                MOV     BCNT,8.(R0)     ;TOTAL COUNT TO WRITE
                                MOV     REC,10.(R0)     ;AT  RECORD N
                                MOV     #RSMSIZ,R1      ;THE PACKET SIZE PLUS+2
                                TST     (R1)+           ;(FLAG AND COUNT) INTO R1
                                MOV     #RSSNSZ,SNDCNT(R5)  ;LOAD THE  SIZE TO SEND
                                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     BCNT,R2         ;GET # OF DATA BYTES
                                CALL    RSVP            ;SEND (AND RETURN TO SCHEDULER)
                                BIT     #BIT3,@R5       ;FLAG BYTE ERROR?
                                BNE     T               ;YES
                                BIC     #BIT12,@R5      ;FLAG FOR LAST PACKET
                        A:      MOV     #TRBUF,R0       ;POINT TO TOP OF BUFFER AGAIN
                                CMP     R2,#128.        ;START DATA PACKET(S)
                                BHI     B               ;BCNT > 128.!
                                MOV     R2,R1           ;BCNT<128.
```

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

                .NLIST
                .NLIST ME
                .LIST
                .ENDM
```

```
                        ;++
                        ;THE SEEK MACRO IMPLIMENTS THE COMPLETE PROTOCOL TO INITIATE A SEEK
                        ;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
                        ;           UNITS TEST REGISTERS FROM SWAPIN
                        ;           TRBUF - BUFFER ADDRESS
                        ;
                        ; OUTPUTS -
                        ;           XSPKNM = # OF PACKETS EXPECTED
                        ;           XSFLG = FLAG BYTE OF 1ST PACKET
                        ;           XSCNT = BYTE COUNT OF 1ST PACKET
                        ;             . ***
                        ;             .   *  SUBSFQUENT XSFLGS
                        ;             .    >
                        ;             .   *  AND XSCNTS
                        ;             . ***
                        ;
                        ;--

                                .MACRO  TUSEEK  REC,DR,?A

                                .NLIST
                                .LIST ME
                                .LIST

                        A:      MOV     #TRBUF,R0       ;-->(POINT TO) XMIT BUFFER
                                MOVB    #RSCMND,@R0     ;FORM COMMAND MESSAGE PACK
                                MOVB    #RSMSIZ,1(R0)   ;THIS BIG
                                MOVB    #RSSSEK,2(R0)   ;OP CODE IS SEEK
                                MOV     REC,10.(R0)     ;TO THIS RECORD
                                MOVB    DR,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=# OF BYTES
                                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     A               ;YES
```

```
    1302                                    .NLIST
    1303                                    .NLIST ME
    1304                                    .LIST
    1305                                    .ENDM
```

```
1308                    ;++
1309                    ;THE RETRY MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO INITIATE
1310                    ;A RETRY (READ OPERATION) SEQUENCE.
1311                    ;
1312                    ;SETS UP THE EXPECTED PROTOCOL RESPONSES:  THE NUMBER OF PACKETS
1313                    ;(XSPKNM) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT).  CALLS
1314                    ;'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
1315                    ;CHECKSUM.
1316                    ;
1317                    ; INPUTS  - DEVICE BLOCK @R5
1318                    ;           TRBUF - BUFFER ADDRESS
1319                    ;           UNITS TEST REGISTERS FROM SWAPIN
1320                    ;
1321                    ; OUTPUTS - SNDCNT(R5) = # OF BYTES TO SEND
1322                    ;           XSPKNM = # OF PACKETS EXPECTED
1323                    ;           XSFLG = FLAG BYTE OF 1ST PACKET
1324                    ;           XSCNT = BYTE COUNT OF 1ST PACKET
1325                    ;             . ***
1326                    ;             .   *  SUBSEQUENT XSFLGS
1327                    ;             .    >
1328                    ;             .   *  AND XSCNTS
1329                    ;             . ***
1330                    ;--
1331
1332                            .MACRO  TURTRY  REC,BCNT,DR,?A,?B,?C,?D,?E
1333
1334                            .NLIST
1335                            .LIST ME
1336                            .LIST
1337
1338                    D:      MOV     #TRBUF,R0       ;FORM CMND PACK:
1339                            MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
1340                            MOVB    #RSMSIZ,1(R0)   ;THIS BIG
1341                            MOVB    #RSSRD,2(R0)    ;OP CODE-READ
1342                            MOV     REC,10.(R0)     ;THIS RECORD
1343                            MOVB    DR,4.(R0)       ;THIS DRIVE
1344                            CLRB    3(R0)           ;PRESET NORM THRESHOLD
1345                            TSTB    @R5             ;REDUCED?
1346                            BPL     E               ;NO
1347                            INCB    3(R0)           ;YES-CHANGE THRESHOLD
1348                    E:      MOV     BCNT,8.(R0)     ;# BYTES DESIRED
1349                            MOVB    #020,5.(R0)     ;MAINTENANCE MODE
1350                            CLR     6.(R0)          ;NO SEQUENCE #
1351                            MOV     #RSMSIZ,R1      ;SIZE OF PACKET
1352                            TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
1353                            MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND
1354
1355                            CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
1356                            MOV     R1,(R0)         ;INSERT IN PACKET
1357
1358                            MOV     BCNT,R1         ;SET EXPECTATIONS:
1359                                                    ;CALC # OF DATA PACKETS TO EXPECT
1360                            MOV     #XSFLG,R3       ;OFFSET OF FLAG
1361                            ADD     R5,R3           ;ABS. ADDR. OF XSFLG
1362                            CLR     R2              ;PRESET
1363                    A:      INC     R2              ;# PACKETS EXPECTED
1364                            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 EXPECT
                                                        MOV     #P.SEND,(R3)+   ;EXPECT AN END
                                                        MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES

                                                        CALL    RSVP            ;SEND
                                                                                ;AND RETURN TO SCHEDULER


                                        .NLIST
                                        .NLIST ME
                                        .LIST
                                        .ENDM
```

```
1383                            ;++
1384                            ;THE READ MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO INITIATE
1385                            ;A READ SEQUENCE.
1386                            ;
1387                            ;SETS UP THE EXPECTED PROTOCOL RESPONSES:  THE NUMBER OF PACKETS
1388                            ;(XSPKNM) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT).  CALLS
1389                            ;'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
1390                            ;CHECKSUM.
1391                            ;
1392                            ; INPUTS  - DEVICE BLOCK @R5
1393                            ;           TRBUF - BUFFER ADDRESS
1394                            ;           UNITS TEST REGISTERS FROM SWAPIN
1395                            ;
1396                            ; OUTPUTS - SNDCNT(R5) = # OF BYTES TO SEND
1397                            ;           XSPKNM = # OF PACKETS EXPECTED
1398                            ;           XSFLG = FLAG BYTE OF 1ST PACKET
1399                            ;           XSCNT = BYTE COUNT OF 1ST PACKET
1400                            ;             . ***
1401                            ;             .   *  SUBSEQUENT XSFLGS
1402                            ;             .    >
1403                            ;             .   *  AND XSCNTS
1404                            ;             . ***
1405                            ;--
1406
1407                                    .MACRO  TUREAD  REC,BCNT,DR,VER,?A,?B,?C,?D,?E
1408
1409                                    .NLIST
1410                                    .LIST ME
1411                                    .LIST
1412
1413
1414                            E:      MOV     #TRBUF,R0       ;FORM CMND PACK:
1415                                    MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
1416                                    MOVB    #RSMSIZ,1(R0)   ;THIS BIG
1417                                    MOVB    #RSSRD,2(R0)    ;OP CODE IS READ
1418                                    MOV     REC,10.(R0)     ;THIS RECORD
1419                                    MOVB    DR,4.(R0)       ;THIS DRIVE
1420                                    MOVB    VER,3.(R0)      ;VERIFY
1421                                    MOV     BCNT,8.(R0)     ;TOTAL BYTES TO READ
1422                                    MOVB    #020,5.(R0)     ;MAINTENANCE MODE
1423                                    CLR     6.(R0)          ;NO SEQUENCE #
1424                                    MOV     #RSMSIZ,R1      ;GET SIZE OF PACKET
1425                                    TST     (R1)+           ;+2 FOR CHECKSUM
1426                                    MOV     #RSSNSZ,SNDCNT(R5)  ;SIZE TO SEND
1427                                    CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
1428                                    MOV     R1,(R0)         ;INSERT CHECKSUM
1429
1430                                    MOV     BCNT,R1         ;SET EXPECTATIONS:
1431                                                            ;CALC # OF DATA PACKETS TO EXPECT:
1432                                    MOV     #XSFLG,R3       ;GET OFFSET
1433                                    ADD     R5,R3           ;ABS. ADDR. OF XSFLG
1434                                    CLR     R2              ;PRESET AS NONE
1435                            A:      INC     R2              ;# PACKETS EXPECTED
1436                                    MOV     #RSDATA,(R3)+   ;LOAD XSFLG
1437                                    MOV     #132.,(R3)+     ;AND EXPECTED COUNT
1438                                    SUB     #128.,R1        ;NEG RESULT LAST TIME
1439                                    BLOS    C               ;LAST TIME
```

```
1440                                                    BR      A               ;MORE TO DO
1441                                            C:      INC     R2              ;ADD ONE FOR END PACK
1442                                                    MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
1443                                                    MOV     #RSEND,(R3)+    ;EXPECT AN END ALSO...
1444                                                    MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES
1445                                                    CALL    RSVP            ;SEND
1446                                                                            ;AND RETURN TO SCHEDULER
1447                                            D:      BIT     #BIT10!BIT3,@R5 ;RETRY?
1448                                                    BEQ     B               ;NO.
1449                                                    TURTRY  REC,BCNT,DR     ;YES
1450                                                    BR      D               ;ANOTHER RETRY?
1451                                            B:      NOP                     ;NO
1452
1453
1454                                    .NLIST
1455                                    .NLIST ME
1456                                    .LIST
1457                                    .ENDM
```

```
                        ;++
                        ;THE SELF TEST MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO
                        ;INITIATE A 'DIAGNOSE' 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 REGISTERS TEST 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  TUSELF  ?A

                                .NLIST
                                .LIST ME
                                .LIST

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

                                .NLIST
                                .NLIST ME
                                .LIST
                                .ENDM
```

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

                                .MACRO  TSTID   ADDRES,?A

                                .NLIST
                                .LIST ME
                                .LIST
                                                MOV     ADDRES,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     A               ;BR NO 2ND DRVS
                                                CALL    SETUP           ;REINIT UNITS TSTPC
                                                CALL    RUN             ;REPEAT TEST
                                A:                                      ;DONE
                                .NLIST
                                .NLIST ME
                                .LIST
                                .ENDM
                        ;-------------------------------------------------------------------
```

```
                                        .SBTTL  GLOBAL SUBROUTINES SECTION

                                        ;++
                                        ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES THAT ARE USED
                                        ; TO LINK THE DIAGNOSTIC TO THE SUPERVISOR (THROUGH THE TSTID MACRO).
                                        ;--

                                        ;++
                                        ; SWAPDR
                                        ;    SUBROUTINE TO DETERMINE IF TO TEST OTHER DRIVE (FOR ALL UNITS)
                                        ; INPUTS:       DR(R5) - DRIVE CONFIGURATION
                                        ;               BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
                                        ;               LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
                                        ;
                                        ; OUTPUTS:      DR(R5) UPDATED TO TEST SAME OR OTHER DRIVE
                                        ;               CARRY SET IF SECOND PASS NECESSARY
                                        ;--

005530  005002                  SWAPDR:: CLR     R2              ;FOR # OF DRIVE 1'S.
005532  012737  003350  005630          MOV     #BLKTBL,SWPTR   ;TABLE ADDR. OF 1ST UNIT
005540  017705  000064          1$:     MOV     @SWPTR,R5       ;GET DATA BLOCK ADDR.
005544  032715  100000                  BIT     #BIT15,@R5      ;ABORTED?
005550  001013                          BNE     3$              ;YES
005552  032765  000001  000060          BIT     #BIT0,DR(R5)    ;DID DR. 0?
005560  001007                          BNE     3$              ;NO, DID DR.1 1ST PASS
005562  032765  001000  000060          BIT     #BIT9,DR(R5)    ;YES; 1 SELECTED?
005570  001403                          BEQ     3$              ;NO, ALL DONE
005572  105265  000060                  INCB    DR(R5)          ;YES, SWAP
005576  005202                          INC     R2              ;ONE MORE TO TEST
005600  023727  005630  003366  3$:     CMP     SWPTR,#LSTDEV   ;LAST DEVICE?
005606  103004                          BHIS    4$              ;YES
005610  062737  000002  005630          ADD     #2,SWPTR        ;NO-POINT NEXT
005616  000750                          BR      1$              ;DO

005620  005702                  4$:     TST     R2              ;(CLEAR CARRY),MORE TO DO?
005622  001401                          BEQ     5$              ;NO
005624  000261                          SEC                     ;YES
005626  000207                  5$:     RETURN                  ;RETURN

005630  000000                  SWPTR:  .WORD
```

```
1647                                    ;++
1648                                    ; SETDR - SUBROUTINE TO GET DRIVE FOR 1ST PASS FOR EACH TEST
1649                                    ;
1650                                    ; INPUTS:       DR(R5) - DRIVE CONFIGURATION
1651                                    ;               BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
1652                                    ;               LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
1653                                    ;
1654                                    ; OUTPUTS:      DR(R5) IS SET TO TEST DRIVE 0 OR DRIVE 1
1655                                    ;--
1656
1657
1658 005532  012737  003350  005706  SETDR:: MOV     #BLKTBL,SETPTR  ;TABLE OF ADDR. 1ST UNIT
1659 005540  017705  000042          1$:     MOV     @SETPTR,R5      ;GET DATA BLOCK ADDR.
1660 005544  105065  000060                  CLRB    DR(R5)          ;PRESET AS DR0
1661 005550  032765  000400  000060          BIT     #BIT8,DR(R5)    ;DO DR0?
1662 005556  001002                          BNE     2$              ;YES
1663 005560  105265  000060                  INCB    DR(R5)          ;NO-USE DRIVE 1
1664 005564  023727  005706  003366  2$:     CMP     SETPTR,#LSTDEV  ;MORE UNITS
1665 005572  103004                          BHIS    3$              ;NO-EXIT
1666 005574  062737  000002  005706          ADD     #2,SETPTR       ;YES-GET TABLE ENTRY
1667 005702  000756                          BR      1$              ;CONFIGURE THAT UNIT
1668 005704  000207                  3$:     RETURN
1669 005706  000000                  SETPTR: .WORD
```

```
                                        ;++
                                        ; CLRALL - CLEARS INPUT BUFFER FOR RESPONSE FROM UNIT.
                                        ;
                                        ; INPUTS:       BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
                                        ;               LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
                                        ;
                                        ; OUTPUTS:      ALL UNITS BUFFERS CLEARED.
                                        ;
                                        ; CALLS:        CLRBUF
                                        ;--

005710  012737  003350  006002  CLRALL:: MOV     #BLKTBL,CLRPTR  ;TOP OF TABLE OF ADDRESSES
005716  017705  000060          1$:     MOV     @CLRPTR,R5      ;GET DATA BLOCK
005722  004737  005750                  CALL    CLRBUF          ;CLEAR IT'S RECEIVE BUFFER
005726  023727  006002  003366          CMP     CLRPTR,#LSTDEV  ;LAST DEV?
005734  103004                          BHIS    2$              ;YES
005736  062737  000002  006002          ADD     #2,CLRPTR       ;-->NEXT
005744  000764                          BR      1$              ;CONTINUE
005746  000207                  2$:     RETURN
```

```
                        ;++
                        ; CLRBUF - CLEARS 1 UNIT'S INPUT BUFFER.
                        ; INPUTS:  RCVBUF(R5) IS BUFFER START
                        ;          RCBFSZ - SIZE OF RECEIVE BUFFER IN BYTES
                        ;          RCBFSZ IS SIZE OF BUFFER
                        ; OUTPUTS:  CLEARED AREA.
                        ;--

005750                  CLRBUF:: PUSH   R0              ;SAVE R0
005750  010046                                          MOV     R0,-(SP)

005752                          PUSH    R4              ;SAVE R4
005752  010446                                          MOV     R4,-(SP)

005754  016500  000102          MOV     RCVBUF(R5),R0   ;GET ADDRESS OF BUFFER
005760  012704  001036          MOV     #RCBFSZ,R4      ;SIZE IN BYTES
005764  005020          1$:     CLR     (R0)+           ;CLEAR IT
005766  162704  000002          SUB     #2,R4           ;2 BYTES LESS
005772  001374                  BNE     1$              ;MORE
005774                          POP     R4              ;RESTORE
005774  012604                                          MOV     (SP)+,R4

005776                          POP     R0              ;
005776  012600                                          MOV     (SP)+,R0

006000  000207                  RETURN                  ;EXIT
006002  000000          CLRPTR: .WORD
```

```
                                        ;++
                                        ; SETUP - CALLED WITHIN EACH TEST TO INSERT BEGINNING ADDRESS OF THE
                                        ;         TEST INTO ALL UNITS TEST PC'S.
                                        ; INPUTS:       TSTTOP LOADED WITH TEST ALGORITHMS STARTING ADDR.
                                        ;               BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
                                        ;               LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
                                        ;
                                        ; OUTPUTS:      TSTPC(R5) FOR ALL UNITS
                                        ;               DONE - CLEARED
                                        ;--

006004  005037  003322          SETUP:: CLR     DONE            ;NOT DONE YET
006010  012737  003350  003324          MOV     #BLKTBL,IDPTR   ;TABLE TOP ADDR
006016  017705  175302          1$:     MOV     @IDPTR,R5       ;DEVICE'S DATA BLOCK
006022  013765  003326  000020          MOV     TSTTOP,TSTPC(R5);INSERT PC FOR TOP OF TEST
006030  023727  003324  003366          CMP     IDPTR,#LSTDEV   ;ALL UNITS SET?
006036  103004                          BHIS    2$              ;YES
006040  062737  000002  003324          ADD     #2,IDPTR        ;NO,GET NEXT POINTER
006046  000763                          BR      1$              ;SET HIM UP
006050  000207                  2$:     RETURN                  ;DONE
```

```
                                        ;++
                                        ; RUN - IMPLEMENTS THE CALLS TO SEND PACKETS, RECEIVE PACKETS, THEN
                                        ;       CHECK ANSWERS DURING TEST RUN TIME.
                                        ;
                                        ; INPUTS:  DONE
                                        ; OUTPUTS: NONE
                                        ;--

006052  004737  006102          RUN::   CALL    NXTST           ;MAKE AND SEND NEXT PACK TO ALL
                                                                ;UNABORTED UNITS
006056  005737  003322                  TST     DONE            ;COMPLETE?
006062  001006                          BNE     2$              ;YES
006064  004737  007152                  CALL    GETANS          ;NO,GET ALL RESPONSES

006070                                  BREAK                   ;SUPERVISOR CHECK
006070  104422                                                          TRAP    C$BRK

006072  004737  010576                  CALL    CHKANS          ;CHECK ALL RESPONSES
006076  000765                          BR      RUN             ;CONTINUE TILL DONE
006100  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
                                        ;--

006102  000240                  NXTST:: NOP                     ;
006104  012737  003350  003312          MOV     #BLKTBL,DEVPTR  ;UNIT 0 TO START
006112  017705  175174          1$:     MOV     @DEVPTR,R5      ;GET DATA BLOCK
006116  005715                          TST     @R5             ;ABORTED?
006120  100504                          BMI     2$              ; YES... CHECK NEXT UNIT
006122  032715  000010          3$:     BIT     #BIT3,@R5       ;NO-RETRY 'BAD FLAG'?
006126  001040                          BNE     5$              ;YES...(SEND ABORT;THEN CMD PACK)
006130  032715  020000                  BIT     #BIT13,@R5      ;NO-RETRYING STILL (NO END PACK YET)?
006134  001426                          BEQ     7$              ;NO...
006136  032715  000400                  BIT     #BIT8,@R5       ;RETRYING A WRITE?
006142  001453                          BEQ     4$              ;NO...
006144                                  SWAPIN                  ;YES-GET DEVICE REGESTERS
006144  016500  000006                  MOV     6.(R5),R0
006150  016501  000010                  MOV     8.(R5),R1
006154  016502  000012                  MOV     10.(R5),R2
006160  016503  000014                  MOV     12.(R5),R3
006164  016504  000016                  MOV     14.(R5),R4

006170  020265  000206                  CMP     R2,SAVCNT(R5)   ;CURRENT COUNT = SAVED COUNT? (WHERE WE STARTED)
006174  001036                          BNE     4$              ;NO...(CONTINUE SENDING DATA PACKS)
006176  042737  000004  003306          BIC     #BIT2,SYSTAT    ;YES-CLEAR RETRY FLAGS
006204  042715  020000                  BIC     #BIT13,@R5      ;
006210  000450                          BR      2$              ;CHECK NEXT UNIT.
006212  032715  002000          7$:     BIT     #BIT10,@R5      ;NO-RETRY DATA ERROR?
006216  001445                          BEQ     2$              ;NO...ON TO NEXT UNIT
006220  052737  000002  003306          BIS     #BIT1,SYSTAT    ;SET RETRY STATUS TO 'DATA ERROR' TYPE
006226  000424                          BR      6$              ;YES...

006230                          5$:     SWAPIN                  ;GET DEVICE REGISTERS
006230  016500  000006                  MOV     6.(R5),R0
006234  016501  000010                  MOV     8.(R5),R1
006240  016502  000012                  MOV     10.(R5),R2
006244  016503  000014                  MOV     12.(R5),R3
006250  016504  000016                  MOV     14.(R5),R4

006254  010265  000206                  MOV     R2,SAVCNT(R5)   ;SAVE THE BYTE COUNT (FOR WRITE OPERATION)
                                                                ;TO MARK HOW MANY DATA PACKS TO SEND
006260  004737  014010                  CALL    DOBRK           ;SEND INIT
```

```
1801 006264  032715  100000                    BIT     #BIT15,@R5      ;ABORTED?
1802 006270  001020                            BNE     2$              ;YES...
1803 006272  052737  000004  003306  4$:       BIS     #BIT2,SYSTAT    ;NOT ABORTED-SET RETRY STATUS
1804 006300                          6$:       SWAPIN                  ;GET DEVICE REGISTERS
     006300  016500  000006                                            MOV     6.(R5),R0
     006304  016501  000010                                            MOV     8.(R5),R1
     006310  016502  000012                                            MOV     10.(R5),R2
     006314  016503  000014                                            MOV     12.(R5),R3
     006320  016504  000016                                            MOV     14.(R5),R4

1805 006324  004775  000020                    JSR     PC,@TSTPC(R5)   ;DO TEST FOR
1806 006330  000477                            BR      NXTRET          ;THIS UNIT ONLY-EXIT
1807 006332  023727  003312  003366  2$:       CMP     DEVPTR,#LSTDEV  ;TRY NEXT UNIT?
1808 006340  103004                            BHIS    NXTST2          ;NO
1809 006342  062737  000002  003312            ADD     #2.,DEVPTR      ;YES,->NEXT
1810 006350  000660                            BR      1$              ;GET BLOCK
1811
1812 006352  005037  006532          NXTST2:   CLR     ABONM           ;HERE=NO RETRIES TO DO, NO UNIT ABORTED YET
1813 006356  012737  003350  003312            MOV     #BLKTBL,DEVPTR  ;-->UNIT 0 STORAGE BLOCK
1814 006364  017705  174722          PERDEV:   MOV     @DEVPTR,R5      ;R5-->NEXT DEVICE STORAGE BLOCK
1815
1816 006370  005715                  3$:       TST     @R5             ;ABORTED?
1817 006372  100426                            BMI     4$              ;YES
1818 006374  032715  040000                    BIT     #BIT14,@R5      ;SEND BREAK?
1819 006400  001407                            BEQ     6$              ;NO
1820 006402  004737  014010                    CALL    DOBRK           ;YES
1821 006406  032715  040000                    BIT     #BIT14,@R5      ;SUCCESSFUL INIT?
1822 006412  001016                            BNE     4$              ;NO ON TO NEXT UNIT
1823 006414  005715                            TST     @R5             ;ABORTED?
1824 006416  100414                            BMI     4$              ;YES-ON TO NEXT UNIT
1825 006420                          6$:       SWAPIN                  ;NO,GET DEVICE REGISTERS R0-R4 CONTAINING TEST PARAMETERS
     006420  016500  000006                                            MOV     6.(R5),R0
     006424  016501  000010                                            MOV     8.(R5),R1
     006430  016502  000012                                            MOV     10.(R5),R2
     006434  016503  000014                                            MOV     12.(R5),R3
     006440  016504  000016                                            MOV     14.(R5),R4

1826 006444  004775  000020                    JSR     PC,@TSTPC(R5)   ;INITIATE 1 PACKET TRANSMISSION AND RETURN
1827 006450  005715                  4$:       TST     @R5             ;ABORTED?
1828 006452  100002                            BPL     8$              ;NO-ON TO NEXT UNIT
1829 006454  005237  006532                    INC     ABONM           ;YES...ONE MORE TALLIED
1830 006460  023727  003312  003366  8$:       CMP     DEVPTR,#LSTDEV  ;ALL TU S TRIED?
1331 006466  103004                            BHIS    5$              ;YES
1832 006470  062737  000002  003312            ADD     #2.,DEVPTR      ;NO THE ADDRESS+2=NEXT ADDRESS
1833 006476  000732                            BR      PERDEV          ;DO NEXT UNIT
1834 006500  022737  000010  006532  5$:       CMP     #8.,ABONM       ;ALL ABORTED?
1835 006506  001010                            BNE     NXTRET          ;NO
1836 006510                                    ERRSF   100.,NOMOR      ;YES!
     006510  104454                                                    TRAP    C$ERSF
     006512  000144                                                    .WORD   100
     006514  006534                                                    .WORD   NOMOR
     006516  000000                                                    .WORD   0
1837 006520                          11$:      BREAK                   ;SUPERVISOR BREAK
     006520  104422                                                    TRAP    C$BRK
1838 006522  005237  003340                    INC     ALLGON          ;SET DON'T-PRINT STATISTICS FLAG
1839 006526                                    DOCLN                   ;EXIT
     006526  104444                                                    TRAP    C$DCLN
```

```
   1840 006530  000207                    NXTRET: RETURN
   1841
   1842 006532  000000                    ABONM:  .WORD                          ;THE NUMBER OF ABORTED UNITS
   1843 006534     101      114     114   NOMOR:  .ASCIZ  /ALL UNITS ABORTED!/
   1844                                            .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 DO 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.
                        ;--

006560  000240         RSVP::  NOP                     ;FINISH TEST
006562  012665  000020         MOV     (SP)+,TSTPC(R5) ;SAVE WHERE YOU WERE IN TEST BODY AND
006566                         SWAPOW                  ;SAVE TEST REGISTERS
006566  010065  000006                                 MOV     R0,6.(R5)
006572  010165  000010                                 MOV     R1,8.(R5)
006576  010265  000012                                 MOV     R2,10.(R5)
006602  010365  000014                                 MOV     R3,12.(R5)
006606  010465  000016                                 MOV     R4,14.(R5)

                                                        ;CORRECT FOR RETURN TO SCHEDULER
006612  022737  000002  003342         CMP     #2,TEST9        ;***** IS THIS TEST 9
006620  001007                         BNE     XFNSND          ;***** NO
006622  022765  000000  000210         CMP     #0,MRSP(R5)     ;***** IF SO, IS THIS UNIT MODIFIED
006630  001523                         BEQ     ENDRSP          ;***** YES
006632  012700  027722         NOXOFF: MOV     #TRBUF,R0       ;FOR NORMAL PACKET SEND
006636  000404                         BR      SND             ;SEND XOFF+PACKET
006640  012700  027721         XFNSND: MOV     #TRBUF-1,R0     ;POINT TO XOFF
006644  005265  000070                 INC     SNDCNT(R5)      ;ONE MORE TO SEND, TOO.
006650  004737  007102         SND:    CALL    SNDBYT          ;SEND BYTE
006654  005715                         TST     @R5             ;R5--> TO STATUS BLK
006656  100510                         BMI     6$              ;ABORTED? YES...QUIT
006660  005365  000070                 DEC     SNDCNT(R5)      ;NO, SEND MORE
```

```
1897 006664  001371                        BNE     SND                 ;IF MORE TO SEND
1898 006666  012700  027722                MOV     #TRBUF,R0           ;-->BUFFER
1899 006672  016537  000064  003332        MOV     REC(R5),BLKER       ;PREPARE FOR RECEIVE
1900 006700  156565  000032  000033        BISB    XSPKNM(R5),XSPKNM+1(R5)  ;REPLICATE LO. BYTE TO HI FOR GTPAKS, CHKANS
1901 006706  005065  000076                CLR     SUCCS(R5)           ;NO SUCCESS YET
1902 006712  042715  001000                BIC     #BIT9,@R5           ;NO DATA CHK ERROR YET
1903 006716  016565  000102  000104        MOV     RCVBUF(R5),PKPTR(R5)  ;TOP OF RCV BUFFER GOES THE 1ST PACKET
1904 006724  012704  000034                MOV     #XSFLG,R4           ;FORM
1905 006730  060504                        ADD     R5,R4               ;ADDRESS
1906 006732  010465  000106                MOV     R4,XSPTR(R5)        ;OF 1ST XSFLG
1907
1908 006736  042715  000020                BIC     #BIT4,@R5           ;PRESET AS DATA PAK
1909 006742  121027  000002                CMPB    @R0,#RSCMND         ;WAS IT COMMAND PAK?
1910 006746  001054                        BNE     6$                  ;NO...
1911 006750  116065  000002  000100        MOVB    2(R0),CMDSNT(R5)    ;YES-SAVE COMMAND
1912 006756  052715  000020                BIS     #BIT4,@R5           ;ITS CMND PAK
1913
1914 006762  032715  002000                BIT     #BIT10,@R5          ;RETRYING?
1915 006766  001044                        BNE     6$                  ;YES-DON'T UPDATE ANY STATS OR CONDITION
1916 006770  126027  000002  000002        CMPB    2(R0),#RSSRD        ;NO,A READ?
1917 006776  001012                        BNE     4$                  ;NO
1918 007000  042715  000400                BIC     #BIT8,@R5           ;(FOR HARD/SOFT LOGGING) RD/WR FLAG=0
1919 007004  004737  01364C                CALL    WHCHDR              ;GET DRIVE
1920 007010  103403                        BCS     8$                  ;
1921 007012  005265  000114                INC     RDN0(R5)            ;DRIVE 0
1922 007016  000402                        BR      4$                  ;
1923 007020  005265  000116        8$:     INC     RDN1(R5)            ;DRIVE 1
1924
1925 007024  126027  000002  000003  4$:   CMPB    2(R0),#RSSWR        ;A WRITE?
1926 007032  001022                        BNE     6$                  ;NO
1927 007034  052715  000400                BIS     #BIT8,@R5           ;YES, RD/WR FLAG=1
1928 007040  105760  000003                TSTB    3(R0)               ;VERIFY TOO?
1929 007044  001403                        BEQ     21$                 ;NO
1930 007046  052715  000040                BIS     #BIT5,@R5           ;YES-SET VERIFY FLAG
1931 007052  000402                        BR      22$
1932 007054  042715  000040        21$:    BIC     #BIT5,@R5           ;(NO)-RESET VERIFY FLAG
1933 007060  004737  013640        22$:    CALL    WHCHDR              ;GET DRIVE NO
1934 007064  103403                        BCS     5$                  ;CARRY=DR1
1935 007066  005265  000110                INC     WRTN0(R5)           ;# BLKS WRITTEN DR0
1936 007072  000402                        BR      6$                  ;EXIT
1937
1938 007074  005265  000112        5$:     INC     WRTN1(R5)           ;# BLKS WRITTEN DRV1
1939 007100                        6$:
1940 007100  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
                                ;--

007102                          SNDBYT:: PUSH   R1              ;ENTER R0-->BYTE
007102  010146                                                  MOV     R1,-(SP)


007104  013701  003344          4$:     MOV     CSNRDY,R1       ;GET TIMEOUT CONSTANT FOR NOT READY ERROR
007110  105775  000026          1$:     TSTB    @XMSR(R5)       ;READY TO SEND?
007114  100412                          BMI     2$              ;YES
007116                                  PUSH    R0              ;NO, SAVE R0
007116  010046                                                  MOV     R0,-(SP)


007120                                  BREAK                   ;MONITOR BREAK
007120  104422                                                                          TRAP    C$BRK
007122                                  POP     R0              ;RESTORE
007122  012600                                                  MOV     (SP)+,R0

                                
007124  005301                          DEC     R1              ;ABORTED?
007126  001370                          BNE     1$              ;NO
007130  012704  000056                  MOV     #TOSNDB,R4      ;YES,SET CODE FOR TIMEOUT ERROR
007134  004737  012634                  CALL    LOG             ;LOG IT
007140  000402                          BR      3$              ;QUIT
007142  112075  000030          2$:     MOVB    (R0)+,@XMDB(R5) ;SEND IT
007146                          3$:     POP     R1              ;RESTORE
007146  012601                                                  MOV     (SP)+,R1

007150  000207                          RETURN                  ;DONE
```

```
                                        .SBTTL  GETANS / GETS RESPONSES ROUND ROBIN USING ''XON''

                                        ;++
                                        ; GETANS - IF A UNIT IS RETRYING CLEAR HIS RECEIVE BUFFER (CLRBUF) AND GET
                                        ;          HIS RESPONSE (GTPKS1), ELSE, CLEAR ALL BUFFERS (CLRALL) AND
                                        ;          GET ALL RESPONSES (GTPKS8).
                                        ; INPUTS:  SYSTAT - SYSTEM STATUS WORD.
                                        ;
                                        ; OUTPUTS: SERVST = -1 IF NO RETRIES.
                                        ;--

007152  000240                  GETANS:: NOP                     ;1 UNIT IF RETRY;  ELSE ALL
007154  032737  000006  003306          BIT     #BIT1!BIT2,SYSTAT       ;RETRY?
007162  001010                          BNE     1$              ;YES
007164  012737  177777  010342          MOV     #-1,SERVST      ;PRESET NO UNITS SERVICED
007172  004737  005710                  CALL    CLRALL          ;CLEAR ALL INPUT BUFFERS
007176  004737  007430                  CALL    GTPKS8          ;GET ALL REPLYS
007202  000404                          BR      2$              ;EXIT
007204  004737  005750          1$:     CALL    CLRBUF          ;RETRY-CLEAR 1 UNIT ONLY
                                                                ;R5->UNIT BY NXTST
007210  004737  007220                  CALL    GTPKS1          ;GET 1 REPLY
007214  000207                  2$:     RETURN                  ;DONE

007216  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
                                ;--

007220  000240                  GTPKS1:: NOP                    ;R5->THE UNIT
007222  012703  000034                   MOV     #XSFLG,R3      ;THE OFFSET VALUE OF FLAG
007226  060503                          ADD     R5,R3          ;FORM THE ABSOLUTE ADDRESS
007230  010301                          MOV     R3,R1          ;R3-->ADDR. OF EXPECTED FLAG
007232  062701  000002                  ADD     #2.,R1         ;R1-->ADDR. OF EXPECTED COUNT
007236  012700  007426                  MOV     #EXON,R0       ;R0=ADDRESS
007242  004737  007102                  CALL    SNDBYT         ;XON THE DEVICE
                                                               ;*** TIME CRITICAL
007246  016500  000102                  MOV     RCVBUF(R5),R0  ;***--> TO THE BUFFER
007252  116502  000033                  MOVB    XSPKNM+1(R5),R2 ;***GET THE # OF PACKETS TO RECEIVE
007256  032702  177400                  BIT     #177400,R2     ;***SIGN UN-EXTEND
007262  011137  003316          1$:     MOV     @R1,RCBCNT     ;***HOW MANY BYTES IT SHOULD BE
007266  011337  003314                  MOV     @R3,RCFLG      ;***WHAT THE FIRST BYTE SHOULD BE
007272  004737  010346                  CALL    GTBYTE         ;***GET THE ALL IMPORTANT FLAG
007276  032715  100000                  BIT     #BIT15,@R5     ;TIMEOUT?
007302  001050                          BNE     4$             ;YES
007304  005300                          DEC     R0             ;-> BYTE RECIEVED
007306  111037  003307                  MOVB    @R0,SYSTAT+1   ;SAVE IT AS FLAG BYTE
007312  121037  003314                  CMPB    @R0,RCFLG      ;1ST BYTE WHAT WAS EXPECTED?
007316  001420                          BEQ     2$             ;YES
007320  121027  000002                  CMPB    @R0,#RSEND     ;NO, WAS IT END PAK?
007324  001006                          BNE     14$            ;NO
007326  012737  000016  003316          MOV     #RSNDSZ,RCBCNT ;YES, USE END SIZE FOR COUNT
007334  012702  000001                  MOV     #1,R2          ;AND ASSUME IT'S LAST PACKET!
007340  000407                          BR      2$             ;CONTINUE RECEIVE
007342  121027  000001          14$:    CMPB    @R0,#RSDATA    ;WAS IT DATA?
007346  001026                          BNE     4$             ;NO,CHKANS MAY FIND INIT...
007350  012737  000204  003316          MOV     #RSDASZ,RCBCNT ;YES, SET FOR DATA PAK SIZE
007356  005202                          INC     R2             ;ONE MORE PACK THAN EXPECTED (END PACK)

007360  005200                  2$:     INC     R0             ;RESTORE TO -> NEXT BYTE
007362  005337  003316          5$:     DEC     RCBCNT         ;THAT'S ONE LESS BYTE TO GO
007366  001411                          BEQ     3$             ;DONE
007370  004737  010346                  CALL    GTBYTE         ;GET REST OF PACKET
007374  005765  000074                  TST     DLV(R5)        ;ERROR
007400  001011                          BNE     4$             ;YES-ALL OVER
007402  032715  100000                  BIT     #BIT15,@R5     ;OR IF ABORTED
007406  001006                          BNE     4$             ;THEN QUIT
007410  000764                          BR      5$             ;CONTINUE RECEIVE

007412  005302                  3$:     DEC     R2             ;ONE LESS PACKET TO GO
007414  001403                          BEQ     4$             ;MORE PACKETS IN TRANSACTION?
                                                               ;YES
```

```
 2056 007416  022121                    CMP     (R1)+,(R1)+     ;POINT TO NEW EXPECTED COUNT
 2057 007420  022323                    CMP     (R3)+,(R3)+     ;AND FLAG,
 2058 007422  000717                    BR      1$              ;AND RECEIVE,
 2059 007424  000207              4$:   RETURN                  ;RETURN
 2060
 2061 007426      020             EXON:  .BYTE   RSXON
 2062 007427      023             EXOFF: .BYTE   RSXOFF
```

```
                                SBTTL  GTPKS8 / GET RESPONSES (NO RETRIES)

                                ;++
                                ; GTPKS8 - IF IN TEST 9 AND THE UNIT IS NOT MODIFIED, SKIP THE REST
                                ;          OF THE ROUTINE.  OTHERWISE;
                                ;          SET ALL ABORTED UNITS SERVICED (SERVST: BIT POSITION).  UNTIL
                                ;          ALL UNITS SERVICED (SERVST=0), IF NO MORE PACKETS, SET UNIT
                                ;          SERVICED, ELSE, GET A FLAG BYTE FROM UNIT, DECREMENTING THE
                                ;          NUMBER OF PACKETS LEFT.  CHECK TO SEE IF EXPECTED FLAG,
                                ;          ADJUST COUNT IF NOT, GET REST OF PACKET.  IF WAS DATA PACK,
                                ;          AND NOT IN TEST 9, SEND "XOFF" TO ENHANCE THROUGHPUT AND GO ON
                                ;          TO NEXT UNIT (IF ANY).  IF IN TEST 9, DO NOT SEND "XOFFF".
                                ; INPUTS:  (IMPLIED)UNITS DATA BLOCK POINTED TO BY R5.  NONE PASSED.
                                ;          RSNDSZ - END PACK SIZE
                                ;          RSDNSZ - DATA + END SIZE
                                ;
                                ; OUTPUTS: SYSTAT - UPPER BYTE=1ST BYTE RECEIVED, CURRENT UNIT
                                ;--

007430  000240                  GTPKS8:: NOP                    ;GET ALL UNITS RESPONSES XOFF IF DATA PAK (THROUGHPUT)
007432  022737  000002  003342          CMP     #2,TEST9        ;***** IS THIS TEST 9
007440  001006                        BNE     1$              ;***** NO
007442  022765  000000  000210        CMP     #0,MRSP(R5)     ;***** IF SO, IS THIS UNIT MODIFIED
007450  001002                        BNE     1$              ;***** YES, CONTINUE NORMALLY
007452  000137  010146                JMP     ENDGP8          ;***** ELSE, SKIP ROUTINE
007456  012737  003350  010344  1$:   MOV     #BLKTBL,GTPTR   ;->1ST
007464  017705  000654          GTAGIN: MOV     @GTPTR,R5       ;GET DATA BLOCK
007470  032715  100000                BIT     #BIT15,@R5      ;ABORTED?
007474  001404                        BEQ     2$              ;NO
007476  004737  010256                CALL    SETSRV          ;YES-SET' SERVICED' AND
007502  000137  010104                JMP     GTDOWN          ;ON TO NEXT UNIT
007506  105765  000033          2$:   TSTB    XSPKNM+1(R5)    ;NO, ANY PACKETS LEFT?
007512  001004                        BNE     3$              ;YES
007514  004737  010256                CALL    SETSRV          ;NO-HE'S DONE
007520  000137  010104                JMP     GTDOWN          ;SO ON TO NEXT UNIT
007524  105365  000033          3$:   DECB    XSPKNM+1(R5)    ;NOW ITS ONE LESS PACKET
007530  017537  000106  003314        MOV     @XSPTR(R5),RCFLG  ;GET EXPECTED FLAG
007536  062765  000002  000106        ADD     #2,XSPTR(R5)    ;--> COUNT
007544  017537  000106  003316        MOV     @XSPTR(R5),RCBCNT  ;AND EXPECTED COUNT
007552  022737  000002  003342        CMP     #2,TEST9        ;***** IF TEST 9
007560  001404                        BEQ     1$              ;***** DO NOT SEND XON
007562  012700  007426                MOV     #EXON,R0        ;-> XON
                                                              ;***TIME CRITICAL
007566  004737  007102                CALL    SNDBYT          ;***SEND IT
007572  016500  000104          1$:   MOV     PKPTR(R5),R0    ;***->WHERE 1ST BYTE GOES
007576  004737  010346                CALL    GTBYTE          ;***GET IT
007602  032715  100000                BIT     #BIT15,@R5      ;ABORTED?
007606  001404                        BEQ     4$              ;NO-CONTINUE
007610  105065  000033                CLRB    XSPKNM+1(R5)    ;YES-NO MORE PACKETS EXPECTED
007614  000137  010104                JMP     GTDOWN          ;ON TO NEXT
007620  005300                  4$:   DEC     R0              ;-->BYTE JUST RECEIVED
007622  111037  003307                MOVB    @R0,SYSTAT+1    ;SAVE IT
007626  121037  003314                CMPB    @R0,RCFLG       ;IS IT WHAT EXPECTED?
007632  001436                        BEQ     GTOK            ;YES
007634  105065  000033          UNXPCT: CLRB    XSPKNM+1(R5)    ;NO, MUST BE LAST REPLY
007640  121027  000002                CMPB    @R0,#RSEND      ;MAYBE AN END PAK?
007644  001004                        BNE     4$              ;NO
```

```
2122 007646  012737  000016  003316          MOV     #RSNDSZ,RCBCNT  ;YES, USE PROPER COUNT
2123 007654  000406                          BR      GTUM            ;AND GET IT
2124 007656  121027  000001          4$:     CMPB    @R0,#RSDATA     ;IS IT DATA?
2125 007662  001110                          BNE     GTDOWN          ;NO, ALL OVER, CHKANS WILL INIT UNIT
2126 007664  012737  000222  003316          MOV     #RSNDSZ,RCBCNT  ;YES, USE COUNT OF DATA + END PAK SURE TO FOLLOW
2127 007672  005200                  GTUM:   INC     R0              ;WHERE TO STUFF THE REST
2128 007674  005337  003316          5$:     DEC     RCBCNT          ;ONE DOWN
2129 007700  001501                          BEQ     GTDOWN          ;NONE TO GO
2130 007702  004737  010346                  CALL    GTBYTE          ;MORE TO GO
2131 007706  032715  100000                  BIT     #BIT15,@R5      ;TIMEOUT?
2132 007712  001074                          BNE     GTDOWN          ;YES
2133 007714  005765  000074                  TST     DLV(R5)         ;BUT DLV ERROR?
2134 007720  001765                          BEQ     5$              ;NO
2135 007722  105065  000033                  CLRB    XSPKNM+1(R5)    ;YES-LAST TIME
2136 007726  000466                          BR      GTDOWN          ;ON TO NEXT
2137
2138 007730  005200                  GTOK:   INC     R0              ;NEXT PLACE IN BUFFER
2139
2140                                  ;*****************************************************************************
2141 007732  022737  000002  003342  1$:     CMP     #2,TEST9        ;*** REV.- IF,   NOT TEST 9
2142 007740  001022                          BNE     7$              ;*** REV.- THEN, NO MRSP HANDSHAKING REQUIRED
2143 007742                                  PUSH    R0              ;*** REV.- ELSE, TEST MRSP HANDSHAKE.
     007742  010046                                                  MOV     R0,-(SP)


2144 007744  012737  000002  010254          MOV     #2,MRSDLY       ;*** REV.- DELAY FOR WAIT LOOP
2145                                                                  ;*** REV.- THIS IS THE BEGINNING DELAY LOOP
2146 007752  005000                  2$:     CLR     R0              ;*** REV.-
2147 007754  005300                  3$:     DEC     R0              ;*** REV.-
2148 007756  001376                          BNE     3$              ;*** REV.-
2149 007760  005337  010254                  DEC     MRSDLY          ;*** REV.-
2150 007764  001372                          BNE     2$              ;*** REV.- THIS IS THE END OF DELAY LOOP
2151
2152 007766  105775  000022                  TSTB    @RCSR(R5)       ;*** REV.- IF,   DONE SET,
2153 007772  001066                          BNE     ERRMOD          ;*** REV.- THEN, IT'S AN ERROR BECAUSE
2154                                                                  ;*** REV.-       THERE WAS NO MRSP HANDSHAKE.
2155 007774  012700  010252                  MOV     #MODRSP,R0      ;*** REV.- ELSE, SEEMS TO BE OK, LETS
2156 010000  004737  007102                  CALL    SNDBYT          ;*** REV.-       SEND A 'CONTINUE' AND
2157 010004                                  POP     R0              ;*** REV.-       SEE IF HANDSHAKE WORKS.
     010004  012600                                                  MOV     (SP)+,R0

2158                                  ;*****************************************************************************
2159 010006  005337  003316          7$:     DEC     RCBCNT          ;MORE BYTES?
2160 010012  001413                          BEQ     4$              ;NO-ALL DONE
2161 010014  004737  010346                  CALL    GTBYTE          ;YES-GET IT
2162 010020  032715  100000                  BIT     #BIT15,@R5      ;TIMEOUT?
2163 010024  001027                          BNE     GTDOWN          ;YES
2164 010026  005765  000074                  TST     DLV(R5)         ;ERROR?
2165 010032  001737                          BEQ     1$              ;NO
2166 010034  105065  000033                  CLRB    XSPKNM+1(R5)    ;LAST TIME
2167 010040  000421                          BR      GTDOWN          ;EXIT
2168 010042  122775  000001  000104  4$:     CMPB    #RSDATA,@PKPTR(R5)  ;WAS DATA?
2169 010050  001015                          BNE     GTDOWN          ;NO, ALL DONE
2170 010052  010065  000104                  MOV     R0,PKPTR(R5)    ;START OF NEXT PACK NEXT TIME
2171                                  ;*****************************************************************************
2172 010056  022737  000002  003342          CMP     #2,TEST9        ;*** REV.- IF,   TEST 9
2173 010064  001003                          BNE     20$             ;*** REV.- ELSE,
```

```
                        TST     MRSP(R5)        ;*** REV.- ANDIF, MRSP
                        BNE     GTDOWN          ;*** REV.- THEN, NO HANDSHAKE
                    ;*****************************************************************
                    20$:    MOV     #EXOFF,R0       ;XOFF AND SEND TO
                        CALL    SNDBYT          ;ENHANCE THROUGHPUT
                    GTDOWN: ADD     #2.,XSPTR(R5)   ;NEXT XSFLG FOR NEXT TRY
                        CMP     GTPTR,#LSTDEV   ;DONE ONE CYCLE ALL UNITS?
                        BHIS    1$              ;YES
                        ADD     #2,GTPTR        ;NEXT UNIT
                        JMP     GTAGIN          ;CONTINUE RECEIVE
                    1$:     TSTB    SERVST          ;DONE SERVICING ALL PAKS
                                                ;FROM ALL UNITS?
                        BEQ     ENDGP8          ;YES
                        JMP     GTPKS8          ;NO, KEEP TRYING
                    ENDGP8: RETURN              ;RETURN

                    ERRMOD: NOP                 ;*** REV.- MRSP ERROR
                        PRINTF  #MESMRS,UNITNO
010152  013746  027366                                          MOV     UNITNO,-(SP)
010156  012746  010200                                          MOV     #MESMRS,-(SP)
010162  012746  000002                                          MOV     #2,-(SP)
010166  010600                                                  MOV     SP,R0
010170  104417                                                  TRAP    C$PNTF
010172  062706  000006                                          ADD     #6,SP
                        RETURN

                    MESMRS: .ASCIZ  !%N%S9%S2%01%S9%S9%AERROR IN MRSP PROTOCOL'
                        .EVEN
                    MODRSP: .BYTE   RSCONT
                        .EVEN
                    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
                                ;--

010256                          SETSRV: PUSH    R5              ;SET UNIT SERVICED
010256  010546                                                  MOV     R5,-(SP)

010260                                  PUSH    R0
010260  010046                                                  MOV     R0,-(SP)

010262  011505                          MOV     @R5,R5          ;GET STAT WD
010264  042705  177770                  BIC     #177770,R5      ;MASK UNIT #
010270  012700  010322                  MOV     #SRVTBL,R0      ;->TOP OF BIT TABLE
010274  005705                  1$:     TST     R5              ;RIGHT ONE?
010276  001404                          BEQ     2$              ;YES
010300  062700  000002                  ADD     #2,R0           ;NO, ->NEXT
010304  005305                          DEC     R5              ;1 LESS
010306  000772                          BR      1$              ;CONTINUE
010310  041037  010342          2$:     BIC     @R0,SERVST      ;MOW IT DOWN
010314                                  POP     R0
010314  012600                                                  MOV     (SP)+,R0

010316                                  POP     R5
010316  012605                                                  MOV     (SP)+,R5

010320  000207                          RETURN                  ;RETURN

010322  000001                  SRVTBL: .WORD   BIT0            ;BIT POSITION LOOKUP TABLE
010324  000002                          .WORD   BIT1
010326  000004                          .WORD   BIT2
010330  000010                          .WORD   BIT3
010332  000020                          .WORD   BIT4
010334  000040                          .WORD   BIT5
010336  000100                          .WORD   BIT6
010340  000200                          .WORD   BIT7

010342  000000                  SERVST: .WORD
010344  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
                                ;          ONE DOES THEN MIGHT AS WELL GET IT AND SEND 'XON' TO GET
                                ;          THE REST OF THE MESSAGE, OTHERWISE, 'BREAK'.  THEN SEND
                                ;          'XON', AND TEST FOR LONG TIMEOUT (A 30 SECOND REWIND).  IF SO,
                                ;          LOG ERROR, OTHERWISE REPEAT THE ABOVE UNTIL READY OR TIME OUT.
                                ;          REMEMBER TO PRESERVE R0 SINCE THE 'BREAK' TRAP CLOBBERS IT.
                                ;
                                ; INPUTS  - R0 POINTS TO INPUT BUFFER
                                ;          - IMPLIED UNITS DATA BLOCK
                                ;          - CSRCVB TIME OUT MULTIPLIER
                                ;
                                ; OUTPUTS - R0 IS INCREMENTED
                                ;          - DLV (R5) NON-ZERO ON INTERFACE ERROR.
                                ;
                                ; ERROR   - TIME OUT ON RECEIVE
                                ;--

010346  005037  010572          GTBYTE:: CLR    GBTMP           ;TIMECUT REGISTER
010352  013704  003346                   MOV    CSRCVB,R4       ;TIMEOUT ERROR CONSTANT (MULTIPLIER)
010356  105775  000022          1$:      TSTB   @RCSR(R5)       ;READY?
010362  100013                           BPL    3$              ;NO
010364  017565  000024  000074         MOV    @RCDB(R5),DLV(R5)  ;GET ERROR + BYTE
010372  116520  000074                   MOVB   DLV(R5),(R0)+   ;COPY BYTE TO BUFFER
010376  005765  000074                   TST    DLV(R5)         ;ERROR?
010402  100472                           BMI    4$              ;YES-EXIT
010404  005065  000074                   CLR    DLV(R5)         ;NO-RESET
010410  000467                           BR     4$              ;AND EXIT
010412  005337  010572          3$:      DEC    GBTMP           ;DEC T.O. CONSTANT
010416  001357                           BNE    1$              ;STILL VALID

                                ;CODE TO SEE ^C DURING LONG SEEK OR REWIND
                                                                ;HERE GBTMP=0
010420  010037  010574                   MOV    R0,GBTMP2       ;R0 MUST BE PRESERVED!
010424  012700  007427                   MOV    #EXOFF,R0       ;QUIET THE DEVICE
010430  004737  007102                   CALL   SNDBYT          ;BY SENDING XOFF
010434  105775  000022          6$:      TSTB   @RCSR(R5)       ;CHARACTER SLOP OVER?
010440  100415                           BMI    5$              ;YES
010442  005337  010572                   DEC    GBTMP           ;NO-WAIT A WHILE
010446  105737  010572                   TSTB   GBTMP           ;DONE WAITING?
010452  001370                           BNE    6$              ;NO
010454                                   BREAK                  ;YES-NO SLOP OVER
010454  104422                                                                  TRAP    C$BRK
010456  012700  007426                   MOV    #EXON,R0        ;START DEVICE TALKING
010462  004737  007102                   CALL   SNDBYT          ;AGAIN
010466  013700  010574                   MOV    GBTMP2,R0       ;RESTORE R0
010472  000426                           BR     7$              ;END KLUGE
010474  013700  010574          5$:      MOV    GBTMP2,R0       ;RESTORE R0
010500  017565  000024  000074         MOV    @RCDB(R5),DLV(R5)  ;GET ERROR + BYTE
010506  116520  000074                   MOVB   DLV(R5),(R0)+   ;COPY BYTE TO BUFFER
010512  005765  000074                   TST    DLV(R5)         ;ERROR?
010516  100403                           BMI    17$             ;YES-EXIT
```

GTBYTE / GET A BYTE FROM UNIT

```
 2294 010520  005065  000074                    CLR     DLV(R5)          ;NO-CLEAR
 2295 010524  000400                            BR      17$              ;EXIT
 2296 010526  010037  010574          17$:      MOV     R0,GBTMP2        ;AGAIN SAVE R0
 2297 010532  012700  007426                    MOV     #EXON,R0         ;RESTORE TO TALKING STATE
 2298 010536  004737  007102                    CALL    SNDBYT           ;BY SENDING 'XON'
 2299 010542  013700  010574                    MOV     GBTMP2,R0        ;RESTORE R0
 2300 010546  000410                            BR      4$               ;DONE
 2301 010550  005037  010572          7$:       CLR      GBTMP
 2302 010554  005304                            DEC     R4               ;TIMEOUT?
 2303 010556  001277                            BNE     1$               ;NO
 2304 010560  012704  000050                    MOV     #TORCVB,R4       ;YES
 2305 010564  004737  012634                    CALL    LOG              ;LOG ERROR.
 2306 010570  000207                  4$:       RETURN                   ;RETURN
 2307 010572  000000                  GBTMP:    .WORD   0
 2308 010574  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.
                                ;--

010576  000240                  CHKANS:: NOP                    ;IF RETRY THEN CHECK ONE
                                                                ;ELSE CHECK ALL
010600  032737  000006  003306          BIT     #BIT1!BIT2,SYSTAT       ;RETRYING?
010606  001403                  BEQ     CHK8                    ;NO DO NORMAL
010610  004737  010706          CALL    CHKPKS                  ;YES DO SINGLE UNIT
                                                                ;R5 -> UNIT
010614  000432                  BR      CHKANR                  ;ALL DONE

010616  012737  003350  010704  CHK8:   MOV     #BLKTBL,CHKPTR  ;YOU KNOW ... TOP OF TABLE
010624  017705  000054          2$:     MOV     @CHKPTR,R5      ;GET UNIT'S BLOCK ADDRESS
010630  032715  100000                  BIT     #BIT15,@R5      ;ABORTED?
010634  001012                  BNE     3$                      ;YES
010636  022737  000002  003342          CMP     #2,TEST9        ;***** IS THIS TEST 9
010644  001004                  BNE     1$                      ;***** NO-CONTINUE NORMALLY
010646  022765  000000  000210          CMP     #0,MRSP(R5)     ;***** IF SO, IS THIS UNIT MODIFIED
010654  001402                  BEQ     3$                      ;***** NO SKIP NEXT INSTR
010656  004737  010706          1$:     CALL    CHKPKS          ;NO, DO THIS GUY
010662  023727  010704  003366  3$:     CMP     CHKPTR,#LSTDEV  ;ALL DONE?
010670  103004                  BHIS    CHKANR                  ;YES
010672  062737  000002  010704          ADD     #2 CHKPTR       ;NO,-->NEXT DEVICE
010700  000751                  BR      2$                      ;DO IT

010702  000207                  CHKANR: RETURN

010704  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
                                ;--

010706  000240          CHKPKS:: NOP                     ;CHECK WHAT WAS RECIEVED
010710  042715  000010          BIC     #BIT3,@R5       ;CLEAR 'BAD FLAG' RETRY BIT
010714  016500  000102          MOV     RCVBUF(R5),R0   ;GET BUFFER ADDR.
010720  116502  000032          MOVB    XSPKNM(R5),R2   ;AND # OF PACKETS EXPECTED
010724  012703  000034          MOV     #XSFLG,R3       ;THE OFFSET VALUE
010730  060503                  ADD     R5,R3           ;R3-->THIS UNIT XSFLG AGAIN
010732  010301                  MOV     R3,R1           ;COPY TO R1
010734  062701  000002          ADD     #2,R1           ;R1-->XSBCNT FOR 1ST PACKET
010740  010065  000104      1$: MOV     R0,PKPTR(R5)    ;POINT TO PACKET
010744  111037  003307          MOVB    @R0,SYSTAT+1    ;SAVE RCV'D BYTE
010750  011137  003316          MOV     @R1,RCBCNT      ;GET COUNT
010754  011337  003314          MOV     @R3,RCFLG       ;AND FLAG
010760  121013                  CMPB    @R0,@R3         ;1ST BYTE=EXPECTED?
010762  001057                  BNE     5$              ;UH OH...
010764  121027  000020          CMPB    @R0,#RSCONT     ;OK, IS IT 1 BYTE?
010770  001534                  BEQ     7$              ;YES...ONTO NEXT PACK
                                                        ;NO, SO > 1 BYTE (NEVER EXPECT INIT!)
010772  013704  003316          MOV     RCBCNT,R4       ;EXPECTED, SO COUNT MUST BE RIGHT
010776  005744                  TST     -(R4)           ;ADJUST FROM RECEIVE COUNT TO COUNT FOR CHECKSUM
011000  004737  013750          CALL    CKCKSM          ;CHECK CHECKSUM
011004  103005                  BCC     2$              ;NO CARRY...NO INCORRECT
011006  012704  000022          MOV     #BDCHK,R4       ;ERROR
011012  004737  012634          CALL    LOG             ;LOG IT
011016  000521                  BR      7$              ;ON TO NEXT PACK
011020  122710  000002      2$: CMPB    #RSEND,(R0)     ;END PAK?
011024  001005                  BNE     3$              ;NO
011026  004737  011302          CALL    CHKEND          ;YES-CHECK
011032  012702  000001          MOV     #1,R2           ;LAST PACKET
011036  000511                  BR      7$              ;AND FALL THROUGH
011040  122710  000001      3$: CMPB    #RSDATA,@R0     ;DATA PAK?
011044  001012                  BNE     4$              ;NO
011046  022737  000001  003342  CMP     #1,TEST9        ;***** IS THIS TEST 9
```

```
                                        BNE     11$             ;***** NO-CONTTINUE NORMALLY
                                        CLR     MRSP(R5)        ;***** CLR MRSP FLAG
                                        BR      12$             ;***** SKIP INSTR
                                11$:    CALL    COMPAR          ;YES-CHECK DATA
                                12$:    BR      7$              ;ALL DONE?
                                4$:     BIS     #BIT3!BIT13,@R5 ;SET 'BAD FLAG' RETRY FLAGS
                                        MOV     #OTL,R4         ;OUT TO LUNCH
                                        TST     DLV(R5)         ;AH,BUT DLV ERROR?
                                        BEQ     20$             ;NO
                                        MOV     #OVRN,R4        ;YES-USE CORRECT ERROR #
                                20$:    CALL    LOG             ;TALLY
                                        BR      8$              ;DONE

                                ;HERE CHECKS UNEXPECTED RESPONSE

                                5$:     CMPB    #RSINIT,@R0     ;INIT?
                                        BNE     6$              ;NO
                                        BIS     #BIT3!BIT13,@R5 ;YES-SET RETRY FLAGS
                                        MOV     #RCINIT,R4      ; WE GOT AN INIT
                                        CALL    LOG             ;TALLY IT
                                        BR      8$              ;DONE
                                6$:     CMPB    #RSDATA,@R0     ;DATA PAK?
                                        BNE     9$              ;NO
                                        MOV     #RSDASZ,R4      ;YES, USE DATA SIZE
                                        TST     -(R4)           ;ADJUST FOR CHKSUM
                                        CALL    CKCKSM          ;AND CHECK
                                        BCS     10$             ;GOOF
                                        CALL    COMPAR          ;OK, HOW'S THE DATA?
                                                                ;EXPECTED END, GOT
                                                                ;DATA + END.
                                        ADD     #RSDASZ,R0      ;POINT TO END PACK
                                        BR      1$              ;CHECK IT, USE SAME XSFLG

                                9$:     CMPB    #RSEND,(R0)     ;END?
                                        BNE     4$              ;NO-OUT TO LUNCH
                                        MOV     #RSSNSZ,R4      ;YES, TOTAL SIZE MINUS
                                        TST     -(R4)           ;TWO (THE CHKSUM)
                                        CALL    CKCKSM          ;CHECK IT
                                        BCS     10$             ;OOPS
                                        CMP     #1,TEST9        ;***** IS THIS TEST 9
                                        BNE     13$             ;***** NO-CONTINUE NORMALLY
                                        MOV     #1,MRSP(R5)     ;***** IF SO, SET THE MRSP FLAG
                                13$:    CALL    CHKEND          ;OK,NOW TEST SUC. CODE

                                        BR      8$              ;ALL DONE

                                10$:    MOV     #BDCHK,R4       ;CHECKSUM ERROR
                                        CALL    LOG
                                        BR      8$              ;EXIT

                                7$:     DEC     R2              ;ANY PACKETS LEFT TO CHECK?
                                        BEQ     8$              ;NO, ALL DONE
                                        ADD     RCBCNT,R0       ;YES, POINT TO NEXT PACKET
                                        CMP     (R1)+,(R1)+     ;POINT TO NEXT EXPECTED COUNT
                                        CMP     (R3)+,(R3)+     ;AND EXPECTED FLAG
                                        BR      1$              ;TRY ANOTHER,THEY'RE SMALL
                                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, LOG
                                ;          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
                                ;--

011302  000240                  CHKEND:: NOP
011304                                   PUSH    R0              ;R0 --> END PAK
011304  010046                           MOV     R0,-(SP)

011306                                   PUSH    R4
011306  010446                           MOV     R4,-(SP)

011310  032737  000006  003306  1$:      BIT     #BIT1!BIT2,SYSTAT       ;RETRYING?
011316  001406                           BEQ     NOREE           ;NO-CHECK NORMALLY
011320  032737  000004  003306           BIT     #BIT2,SYSTAT    ;IS IT BAD FLAG TYPE?
011326  001454                           BEQ     CHKREE          ;NO(DATA TYPE)
011330  042715  020000                   BIC     #BIT13,@R5      ;YES, SO IF END PACK THEN RETRY'S COMPLETE
011334  004737  012320          NOREE:   CALL    CHKSUC          ;CHECK SUCCESS CODE
011340  032715  100000                   BIT     #BIT15,@R5      ;ABORTED?
011344  001402                           BEQ     3$              ;NO,CONTINUE
011346  000137  012024                   JMP     CHKRET          ;YES,EXIT
011352  105765  000077          3$:      TSTB    SUCCS+1(R5)     ;NO; HOW'D WE DO?
011356  001013                           BNE     CHKERR          ;NOT SO GOOD.
011360  032715  000100                   BIT     #BIT6,@R5       ;OK, HOST FIND DATA PAK ERROR?
011364  001002                           BNE     2$              ;YES
011366  000137  012024                   JMP     CHKRET          ;NO
011372  012704  000014          2$:      MOV     #BDCOM,R4       ;YES;  JUST BAD DATA-NO DATACHK ERR
011376  004737  012634                   CALL    LOG             ;BAD DATA IN PACKET
011402  000137  012024                   JMP     CHKRET          ;QUIT
011406  032715  001000          CHKERR:  BIT     #BIT9,@R5       ;BAD SUCCESS; TU DATA CHK ERROR?
011412  001002                           BNE     1$              ;YES
011414  000137  012024                   JMP     CHKRET          ;NO. ALL DONE.
011420  052715  002000          1$:      BIS     #BIT10,@R5      ;YES-START RETRY
011424  012765  000001  000002          MOV     #1,RETRY(R5)    ;CALL IT 1ST
011432                                   PRINTX  #RTRYN,RETRY(R5)  ;** PRINT **
011432  016546  000002                                                   MOV     RETRY(R5),-(SP)
011436  012746  012204                                                   MOV     #RTRYN,-(SP)
011442  012746  000002                                                   MOV     #2,-(SP)
011446  010600                                                           MOV     SP,R0
011450  104415                                                           TRAP    C$PNTX
```

```
     011452  062706  000006                                                  ADD     #6,SP
2513 011456  000562                        BR      CHKRET          ;ALL DONE
2514 011460  004737  012320       CHKREE:  CALL    CHKSUC          ;CHECK SUCCESS CODE
2515 011464  105765  000077                TSTB    SUCCS+1(R5)     ; SUCCESSFUL YET?
2516 011470  001054                        BNE     UNSUC           ;NO, CHECK COUNT
2517 011472                                PRINTX  #RECOV,RETRY(R5)
     011472  016546  000002                                                  MOV     RETRY(R5),-(SP)
     011476  012746  012044                                                  MOV     #RECOV,-(SP)
     011502  012746  000002                                                  MOV     #2,-(SP)
     011506  010600                                                          MOV     SP,R0
     011510  104415                                                          TRAP    C$PNTX
     011512  062706  000006                                                  ADD     #6,SP
2518 011516  105715                        TSTB    (R5)            ;DETERMINE THRESHOLD
2519 011520  100411                        BMI     2$              ;IT'S MODIFIED
2520 011522                                PRINTX  #THRSLO         ;NORMAL
     011522  012746  012124                                                  MOV     #THRSLO,-(SP)
     011526  012746  000001                                                  MOV     #1,-(SP)
     011532  010600                                                          MOV     SP,R0
     011534  104415                                                          TRAP    C$PNTX
     011536  062706  000004                                                  ADD     #4,SP
2521 011542  000410                        BR      3$
2522 011544                       2$:      PRINTX  #THRSHI         ;ENHANCED
     011544  012746  012152                                                  MOV     #THRSHI,-(SP)
     011550  012746  000001                                                  MOV     #1,-(SP)
     011554  010600                                                          MOV     SP,R0
     011556  104415                                                          TRAP    C$PNTX
     011560  062706  000004                                                  ADD     #4,SP
2523 011564  032715  000400       3$:      BIT     #BIT8,@R5       ;WRITE OR READ OPERATION?
2524 011570  001003                        BNE     4$              ;WRITE
2525 011572  012704  000002                MOV     #SFTRD,R4       ;READ
2526 011576  000402                        BR      5$
2527 011600  012704  000004       4$:      MOV     #SFTWR,R4       ;WRITE
2528 011604  004737  012634       5$:      CALL    LOG
2529 011610  005065  000002                CLR     RETRY(R5)       ;RESTORE TO NORMAL STATE
2530 011614  042715  002200                BIC     #BIT10!BIT7,@R5 ;NO RETRY, NORM THRESHOLD
2531 011620  000501                        BR      CHKRET          ;QUIT
2532
2533 011622  000240               UNSUC:   NOP                     ;RETRYING;  SEE IF HARD YET
2534 011624  032715  001000                BIT     #BIT9,@R5       ;TU DATA CHECK ERROR?
2535 011630  001015                        BNE     2$              ;YES
2536 011632                                PRINTB  #RETERR         ;NO-"OTHER-ERROR" ERROR
     011632  012746  012246                                                  MOV     #RETERR,-(SP)
     011636  012746  000001                                                  MOV     #1,-(SP)
     011642  010600                                                          MOV     SP,R0
     011644  104414                                                          TRAP    C$PNTB
     011646  062706  000004                                                  ADD     #4,SP
2537 011652  005065  000002                CLR     RETRY(R5)       ;NO RETRIES
2538 011656  042715  002200                BIC     #BIT10!BIT7,@R5 ;NO RETRY, NORM THRESHOLD
2539 011662  000460                        BR      CHKRET          ;EXIT
2540 011664  023765  003330  000002  2$:   CMP     MXRTRY,RETRY(R5)  ;YES. DID WE GRADUATE TO HARD?
2541 011672  001425                        BEQ     HRD1            ;YES
2542 011674  005265  000002                INC     RETRY(R5)       ;NO.  JUST ANOTHER
2543 011700                                PRINTX  #RTRYN,RETRY(R5)  ;PRINT OUT
     011700  016546  000002                                                  MOV     RETRY(R5),-(SP)
     011704  012746  012204                                                  MOV     #RTRYN,-(SP)
     011710  012746  000002                                                  MOV     #2,-(SP)
     011714  010600                                                          MOV     SP,R0
```

```
                                                                                    TRAP    C$PNTX
                                                                                    ADD     #6,SP
2544 011724 032715 000200                BIT     #BIT7,@R5       ;WAS NORMAL THRESHOLD?
2545 011730 001403                       BEQ     1$              ;YES-REDUCE GAIN
2546 011732 042715 000200                BIC     #BIT7,@R5       ;NO-NORM
2547 011736 000432                       BR      CHKRET
2548 011740 052715 000200          1$:   BIS     #BIT7,@R5       ;REDUCED
2549 011744 000427                       BR      CHKRET          ;DONE
2550 011746 000240                 HRD1: NOP                     ;HERE IS HARD ERROR!
2551 011750                              PRINTX  #UNREC
     011750 012746 012224                                                           MOV     #UNREC,-(SP)
     011754 012746 000001                                                           MOV     #1,-(SP)
     011760 010600                                                                  MOV     SP,R0
     011762 104415                                                                  TRAP    C$PNTX
     011764 062706 000004                                                           ADD     #4,SP
2552 011770 032715 000400                BIT     #BIT8,@R5       ;RD OR WR?
2553 011774 001003                       BNE     4$              ;WRITE
2554 011776 012704 000016                MOV     #HRDRD,R4       ;READ
2555 012002 000402                       BR      5$              ;LOG IT
2556 012004 012704 000020          4$:   MOV     #HRDWR,R4       ;WRITE
2557 012010 004737 012634          5$:   CALL    LOG             ;LOG IT
2558 012014 005065 000002                CLR     RETRY(R5)       ;BACK TO NORMAL
2559 012020 042715 002200                BIC     #BIT10!BIT7,@R5 ;NO RETRY, NOT REDUCED
2560
2561 012024 042737 000006 003306 CHKRET: BIC     #BIT1!BIT2,SYSTAT       ;NO SYSTEM RETRY NEXT PASS
2562 012032 042715 000100                BIC     #BIT6,@R5       ;NO MORE HOST DATA CHECK ERROR
2563 012036                              POP     R4
     012036 012604                                                  MOV     (SP)+,R4

2564 012040                              POP     R0
     012040 012600                                                  MOV     (SP)+,R0

2565 012042 000207                       RETURN
2566
2567
2568 012044      045 101      122 RECOV:  .ASCIZ  /%ARECOVERED FROM DATA CHECK ERROR RETRY # %D1%N/
2569                                     .EVEN
2570 012124      045 101      040 THRSLO: .ASCIZ  /%A NORMAL THRESHOLD%N/
2571                                     .EVEN
2572 012152      045 101      040 THRSHI: .ASCIZ  /%A MODIFIED THRESHOLD %N/
2573                                     .EVEN
2574 012204      045 101      122 RTRYN:  .ASCIZ  /%ARETRY # %D1%N/
2575                                     .EVEN
2576 012224      045 101      125 UNREC:  .ASCIZ  /%AUNRECOVERABLE%N/
2577                                     .EVEN
2578 012246      045 101      117 RETERR: .ASCIZ  /%AOTHER ERROR DURING RETRY : EXIT RETRY%N/
2579                                     .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
                                        ;--

012320  000240                          CHKSUC:: NOP
012322  016065  000002  000076                   MOV     2(R0),SUCCS(R5) ;R0-->END PACKET
012330  122760  000000  000003                   CMPB    #ESOK,3(R0)     ;GET SUCCESS BYTE
012336  001535                                   BEQ     12$             ;COMPLETE SUCCESS-EXIT

012340  122760  000001  000003                   CMPB    #ESTRY,3(R0)    ;OK BUT RETRIES?
012346  001012                                   BNE     20$             ;NO
012350  126527  000100  000002                   CMPB    CMDSNT(R5),#RSSRD;A READ?
012356  001001                                   BNE     22$             ;NO

012360  000520                                   BR      10$             ;NO RETRIES IN MAINTENANCE!
012362  126527  000100  000003  22$:     CMPB    CMDSNT(R5),#RSSWR       ;A WRITE?
012370  001001                                   BNE     20$             ;NO
012372  000513                                   BR      10$             ;LOG IT
012374  122760  177737  000003  20$:     CMPB    #ESNOMO,3(R0)   ;NO MOTOR?
012402  001003                                   BNE     1$              ;NO
012404  012704  000030                           MOV     #NOMOT,R4       ;YES-
012410  000506                                   BR      11$             ;LOG

012412  122760  177757  000003  1$:      CMPB    #ESCKS,3(R0)    ;"DATA CHECK" ERROR?
012420  001003                                   BNE     2$              ;NO
012422  052715  001000                           BIS     #BIT9,@R5       ;SET DATA-CHK-ERROR FLAG
012426  000501                                   BR      12$             ;DONT LOG

012430  126527  000100  000007  2$:      CMPB    CMDSNT(R5),#RSSSLF  ;SELF TEST?
012436  001006                                   BNE     3$              ;NOPE
012440  105760  000003                           TSTB    3(R0)           ;YES, NEG. IF ERROR
012444  100072                                   BPL     12$             ;OK

012446  012704  000044                           MOV     #SLFER,R4       ;YES-ERROR
012452  000465                                   BR      11$             ;LOG IT

012454  122760  177740  000003  3$:      CMPB    #ESSK,3(R0)     ;SEEK ERROR?
012462  001005                                   BNE     4$              ;NO
012464  012704  000024                           MOV     #SKERR,R4       ;YES-
012470  052705  040000                           BIS     #BIT14,R5       ;SET 'DOBRK' FLAG
012474  000454                                   BR      11$             ;LOG

012476  122760  177767  000003  4$:      CMPB    #ESNCRT,3(R0)   ;NO CART?
012504  001003                                   BNE     5$              ;NO
012506  012704  000054                           MOV     #NCART,R4       ;YES-
012512  000445                                   BR      11$             ;LOG
```

```
2639 012514  122760  177720  000003  5$:     CMPB    #ESCMD,3(R0)    ;NO UNDERSTAND HOST?
2640 012522  001003                          BNE     6$              ;NO
2641 012524  012704  000040                  MOV     #CMNDER,R4      ;YES-
2642 012530  000436                          BR      11$             ;LOG
2643
2644 012532  122760  177770  000003  6$:     CMPB    #ESNONX,3(R0)   ,NON EXISTENT UNIT?
2645 012540  001002                          BNE     7$              ;NO
2646 012542  012704  000036                  MOV     #NOUNIT,R4      ;YES-
2647 012546  000427                          BR      11$             ;LOG
2648
2649 012550  122760  177765  000003  7$:     CMPB    #ESWLOC,3(R0)   ;WRITE LOCKED?
2650 012556  001003                          BNE     8$              ;NO
2651 012560  012704  000026                  MOV     #WRLOCK,R4      ;YES-
2652 012564  000420                          BR      11$             ;LOG
2653
2654 012566  122760  177776  000003  8$:     CMPB    #ESPART,3(R0)   ;PARTIAL OP?
2655 012574  001003                          BNE     9$              ;NO
2656 012576  012704  000034                  MOV     #PARTL,R4       ;YES-
2657 012602  000411                          BR      11$             ;LOG
2658
2659 012604  122760  177711  000003  9$:     CMPB    #ESREC,3(R0)    ;WRONG RECORD?
2660 012612  001003                          BNE     10$             ;NO
2661 012614  012704  000042                  MOV     #RECERR,R4      ;YES-
2662 012620  000402                          BR      11$             ;LOG
2663
2664 012622  012704  000046          10$:    MOV     #SUCCTL,R4      ;UNDEFINED
2665 012626  004737  012634          11$:    CALL    LOG             ;LOG ERROR
2666 012632  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
                        ;--

012634                  LOG::   PUSH    R0                      MOV     R0,-(SP)
012634  010046
012636                          PUSH    R1                      MOV     R1,-(SP)
012636  010146
012640                          PUSH    R3                      MOV     R3,-(SP)
012640  010346
012642                          PUSH    R4                      MOV     R4,-(SP)
012642  010446


                        
012644  011537  002074          MOV     @R5,L$LUN       ;GET UNIT NUMBER
012650  042737  177770  002074  BIC     #177770,L$LUN   ;MASK IT OFF
012656  010465  000004          MOV     R4,ABNDX(R5)    ;SAVE INDEX IN CASE OF ABORT MESSAGE
012662  012703  000120          MOV     #LGOFST,R3      ;OFFSET TO LOW ORDER BYTE (DRIVE0)
012666  060403                  ADD     R4,R3           ;FORM INDEX OF PARAM. TO UPDATE
012670  060503                  ADD     R5,R3           ;FORM ABSOLUTE ADDR. THIS UNIT
012672  004737  013640          CALL    WHCHDR          ;SEE WHICH DRIVE T'WAS
012676  103001                  BCC     2$              ;WAS DRIVE 0
012700  005203                  INC     R3              ;DRIVE 1; POINT TO UPPER BYTE
012702  122713  000377  2$:     CMPB    #255.,@R3       ;POTENTIAL OVERFLOW POSSIBLE?
012706  001005                  BNE     LOGOK           ;NO
012710                  LOGO:   ERRDF   0.,OVRFLO,ERRDES ;YES
012710  104455                                                  TRAP    C$ERDF
012712  000000                                                  .WORD   0
012714  013534                                                  .WORD   OVRFLO
012716  013170                                                  .WORD   ERRDES
012720  000512                  BR      ABO             ;ABORT UNIT
012722  105213          LOGOK:  INCB    @R3             ;INCREMENT THE ERROR
012724  111304                  MOVB    @R3,R4          ;TEMP'LY SAVE IT
012726  016503  000004          MOV     ABNDX(R5),R3    ;GET INDEX AGAIN
012732  012701  002226          MOV     #RSNTAB,R1      ;FORM ADRS OF MSG
012736  066501  000004          ADD     ABNDX(R5),R1    ;LIKE THIS
012742  042701  000001          BIC     #BIT0,R1        ;INSURE WORD BOUNDARY
012746  032737  000004  016752  BIT     #EVL,FLGLOC     ;EVL SELECTED?
012754  001414                  BEQ     LOGOK2          ;NO-CONT
```

```
                                        CMPB    EVLTHR,R4       ;YES,OVER THRESHOLD?
                                        BHI     LOGOK2          ;NO
                                        MOV     R3,DFTL1+2      ;YES,LOAD ERROR #
                                        MOV     @R1,DFTL1+4     ;AND MESSAGE ADDR
                                DFTL1:  ERRDF   0,DFTL1,ERRDES  ;ERROR
                                                                        TRAP    C$ERDF
                                                                        .WORD   0
                                                                        .WORD   DFTL1
                                                                        .WORD   ERRDES
                                        BR      ABO             ;DROP IT
                                LOGOK2: CMPB    R3,#BDCOM       ;'NEVER FATAL' TYPE?
                                        BHIS    NTSFT           ;NO
                                        MOV     R3,LOG1+2       ;YES, ERROR CODE
                                        MOV     @R1,LOG1+4      ;DESCRIPTION
                                LOG1:   ERRSOFT 0,,LOG1,ERRDES
                                                                        TRAP    C$ERSOFT
                                                                        .WORD   0
                                                                        .WORD   LOG1
                                                                        .WORD   ERRDES
                                        BR      LOGO            ;EXIT

                                NTSFT:  CMPB    R3,#WRLOCK      ;ONE TRY?
                                        BLO     MABEE           ;NO, MAYBE A MULTIPLE
                                        MOV     R3,LOG2+2,      ;YES
                                        MOV     @R1,LOG2+4
                                LOG2:   ERRHRD  0,LOG2,ERRDES   ;PRINT HARD MESSAGE
                                                                        TRAP    C$ERHRD
                                                                        .WORD   0
                                                                        .WORD   LOG2
                                                                        .WORD   ERRDES
                                        BR      ABO             ;DROP UNIT

                                MABEE:  BIC     #177400,R4      ;NEGATE SIGN EXTEND
                                1$:     SUB     FTLNM,R4        ;SEE IF MULTIPLE OF
                                        BEQ     HRD             ;FTLNM-YES!
                                        BLO     SFT             ;NO
                                        BR      1$              ;NOT THERE YET

                                SFT:    MOV     R3,LOG3+2       ;ERROR CODE
                                        MOV     @R1,LOG3+4      ;DESCRIPTION
                                LOG3:   ERRSOFT 0,LOG3,ERRDES
                                                                        TRAP    C$ERSOFT
                                                                        .WORD   0
                                                                        .WORD   LOG3
                                                                        .WORD   ERRDES
                                        BR      LOGO            ;EXIT
                                HRD:    MOV     R3,LOG3B+2      ;HARD ERROR CODE
                                        MOV     @R1,LOG3B+4     ;DESCRIPTION
                                LOG3B:  ERRHRD  0,LOG3B,ERRDES
                                                                        TRAP    C$ERHRD
                                                                        .WORD   0
                                                                        .WORD   LOG3B
                                                                        .WORD   ERRDES

                                ABO:    MOV     @R5,R0          ;GET UNIT NUMBER
                                        BIC     #177770,R0      ;UN-SIGN EXTEND
                                        DODU    R0              ;USE LOGICAL # TO DROP
```

```
     013154  104451                                     TRAP    C$DODU
2747 013156              LOGO:  POP     R4              ;RESTORE
     013156  012604                                     MOV     (SP)+,R4
2748 013160                     POP     R3
     013160  012603                                     MOV     (SP)+,R3
2749 013162                     POP     R1
     013162  012601                                     MOV     (SP)+,R1
2750 013164                     POP     R0
     013164  012600                                     MOV     (SP)+,R0
2751 013166  000207             RETURN                  ;RETURN
```

```
                        ;++
                        ; ERRDES - CONTAINS CODE FOR EXTENDED ERROR INFORMATION:  DRIVE #,
                        ;          BLOCK #, ETC.
                        ;--

                        ERRDES::
                        BGNMSG  ERRDES          ;ERROR DESCRIPTION
013170                  PUSH    R0
013170  010046                                  MOV     R0,-(SP)

013172                  PUSH    R2
013172  010246                                  MOV     R2,-(SP)

013174  005002          CLR     R2              ;PRESET TO DATA TYPE
013176  032715  000020  BIT     #BIT4,@R5       ;WHAT PACK TYPE?
013202  001401          BEQ     2$              ;DATA
013204  005202          INC     R2              ;COMMAND
013206          2$:     PRINTB  #UNIT,<B,DR(R5)>,R2,<B,SYSTAT+1>
013206  005046                                  CLR     -(SP)
013210  153716  003307                          BISB    SYSTAT+1,(SP)
013214  010246                                  MOV     R2,-(SP)
013216  005046                                  CLR     -(SP)
013220  156516  000060                          BISB    DR(R5),(SP)
013224  012746  013362                          MOV     #UNIT,-(SP)
013230  012746  000004                          MOV     #4,-(SP)
013234  010600                                  MOV     SP,R0
013236  104414                                  TRAP    C$PNTB
013240  062706  000012                          ADD     #12,SP
013244  016500  000064          MOV     REC(R5),R0      ;RECORD NUMBER
013250  016502  000072          MOV     PATTEN(R5),R2   ;DATA EXPECTED
013254                  PRINTB  #RECID,R0,<B,CMDSNT(R5)>,<B,R2>,<B,SUCCS+1(R5)>
013254  005046                                  CLR     -(SP)
013256  156516  000077                          BISB    SUCCS+1(R5),(SP)
013262  005046                                  CLR     -(SP)
013264  150216                                  BISB    R2,(SP)
013266  005046                                  CLR     -(SP)
013270  156516  000100                          BISB    CMDSNT(R5),(SP)
013274  010046                                  MOV     R0,-(SP)
013276  012746  013442                          MOV     #RECID,-(SP)
013302  012746  000005                          MOV     #5,-(SP)
013306  010600                                  MOV     SP,R0
013310  104414                                  TRAP    C$PNTB
013312  062706  000014                          ADD     #14,SP
013316  005765  000074          TST     DLV(R5)         ;DLV ERROR?
013322  001414                  BEQ     3$              ;NO
013324                  PRINTB  #RECID2,DLV(R5) ;YES-PRINT
013324  016546  000074                          MOV     DLV(R5),-(SP)
013330  012746  013616                          MOV     #RECID2,-(SP)
013334  012746  000002                          MOV     #2,-(SP)
013340  010600                                  MOV     SP,R0
013342  104414                                  TRAP    C$PNTB
013344  062706  000006                          ADD     #6,SP
013350  005065  000074          CLR     DLV(R5)         ;RESET
013354          3$:     POP     R2              ;RESTORE
013354  012602                                  MOV     (SP)+,R2
```

```
2775 013356                          POP     R0
     013356  012600                                          MOV     (SP)+,R0

2776 013360                          ENDMSG                  ;EXIT
     013360                                                              L10003:
     013360  104423                                                              TRAP    C$MSG
2777 013362     045     101     104  UNIT::  .ASCIZ  /%ADRIVE#  %01%A  PAK SENT %01%A FLAG RCVD %03%N/
2778                                         .EVEN
2779 013442     045     101     102  RECID:: .ASCIZ  /%ABLOCK#  %04%A COMMAND %02%A EXPCTD %03%A SUCCESS %03%N/
2780                                         .EVEN
2781 013534     103     101     116  OVRFLO: .ASCIZ  /CAN'T UPDATE ERROR OR STATISTIC:OVERFLOW PENDING/
2782                                         .EVEN
2783 013616     045     101     040  RECID2: .ASCIZ  /%A RCDB WAS %06%N/
2784                                         .EVEN
```

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

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

013640  000241                  WHCHDR:: CLC                    ;CLEAR CARRY

013642  105765  000060                 TSTB    DR(R5)         ;DR 0?
013646  001401                         BEQ     2$             ;YES
013650  000261                         SEC                    ;NO

013652  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
                                ;--

013654                          CHKSUM:: PUSH   R3
013654  010346                                  MOV     R3,-(SP)

013656                                   PUSH   R2
013656  010246                                  MOV     R2,-(SP)

013660  042737  000001  003306           BIC    #BIT0,SYSTAT    ;"CHECKSUM IS ODD" BIT
013666  032701  000001                   BIT    #BIT0,R1        ;AN ODD # OF BYTES?
013672  001403                           BEQ    1$              ;NO
013674  052737  000001  003306           BIS    #BIT0,SYSTAT    ;YES

013702  006001                  1$:      ROR    R1              ;/2 FOR WORDS

013704  005003                  2$:      CLR    R3              ;PREP CHECKSUM WORD

013706  062003                  3$:      ADD    (R0)+,R3        ;FORM SUM
013710  005503                           ADC    R3              ;WITH CARRY
013712  005301                           DEC    R1              ;MORE WORDS?
013714  001374                           BNE    3$              ;YES

013716  032737  000001  003306           BIT    #BIT0,SYSTAT    ;WAS IT ODD
013724  001405                           BEQ    4$              ;NO
013726  112002                           MOVB   (R0)+,R2        ;YES GET NEXT BYTE
013730  042702  177400                   BIC    #177400,R2      ;UN-SIGN EXTEND
013734  060203                           ADD    R2,R3           ;ADD IT IN
013736  005503                           ADC    R3              ;AND CARRY JUST IN CASE

013740  010301                  4$:      MOV    R3,R1           ;RETURN IT IN CORRECT PLACE
013742                           POP    R2              ;RESTORE
013742  012602                                  MOV     (SP)+,R2

013744                           POP    R3
013744  0126C3                                  MOV     (SP)+,R3

013746  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
                                ;--


013750                          CKCKSM:: PUSH   R1
013750  010146                                                  MOV     R1,-(SP)

013752                                  PUSH    R0              ;SAVE
013752  010046                                                  MOV     R0,-(SP)

013754  010401                          MOV     R4,R1           ;COPY BYTE COUNT TO CORRECT
013756  004737  013654                  CALL    CHKSUM          ;REGISTER FOR CHKSUM AND
                                                                ;FORM CHECKSUM

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

013762  122001                          CMPB    (R0)+,R1        ;LOWER ORDER CHECK
013764  001005                          BNE     2$              ;WRONG

013766  000301                          SWAB    R1              ;OK-PREP FOR

013770  122001                          CMPB    (R0)+,R1        ;HIGH ORDER CHECK
013772  001002                          BNE     2$              ;WRONG
013774  000241                          CLC                     ;OK-CLEAR SAILING

013776  000401                          BR      3$              ;EXIT

014000  000261                  2$:     SEC                     ;LET ERROR BE KNOWN


014002                          3$:     POP     R0
014002  012600                                                  MOV     (SP)+,R0

014004                                  POP     R1
014004  012601                                                  MOV     (SP)+,R1

014006  000207                          RETURN                  ;RETURN
```

```
                                .SBTTL  DOBRK / MODULE TO INIT TU58 AND TEST INTERRUPTS

                                ;++
                                ; DOBRK - SEND RADIAL SERIAL 'BREAK" TO DEVICE:
                                ;       - SET "BREAK" ON INTERFACE.
                                ;       - SEND 8. NULLS
                                ;       - CLEAR "BREAK" ON INTERFACE
                                ;       - SET VECTORS FOR RCV AND XMIT
                                ;       - SEND 2 BYTES OF "INIT"
                                ;       - RECEIVE "CONTINUE"
                                ;       - IF RECEIVE GARBAGE OR TIMEOUT - ERROR
                                ;       - CLEAR INTERRUPTS AND VECTORS
                                ; INPUTS:  @R5 BIT14 WAS SET - (SEND BREAK)
                                ; OUTPUTS: @R5 BIT14 CLEAR IF SUCCESSFUL INIT.
                                ;          SYSTAT+1 = RECEIVED BYTE
                                ;          ERRORS R4 = ERROR CODE:
                                ;               - SEND NOT READY TIMEOUT (TOSNDB)
                                ;               - NO RESPONSE
                                ;               - DLV ERROR
                                ;               - CAN'T INIT
                                ;--

014010  105037  014543          DOBRK:: CLRB    INITWD+1        ;CLEAR BYTE RECEIVE ADDR
014014  005037  014544                  CLR     BRKTO           ;CLEAR TIME OUT CONSTANT
014020  052775  000001  000026          BIS     #BIT0,@XMSR(R5) ;SET 'BREAK'
014026  012765  000001  000100          MOV     #RSSNIT,CMDSNT(R5)  ;SAY WE SENT 'INIT'
014034  052715  000020                  BIS     #BIT4,@R5       ;PAK SENT TYPE =COMMAND, SORT OF
014040  012704  000010                  MOV     #8.,R4          ;BREAK-IT'S-BACK COUNT=8
014044          1$:     BREAK                   ;SUPERVISOR TAKE FIVE
014044  104422                                                          TRAP    C$BRK
                                                ;FOR ^C CHECK, ETC.
014046  105775  000026                  TSTB    @XMSR(R5)       ;READY?
014052  100410                          BMI     4$              ;YES
014054  005337  014544                  DEC     BRKTO           ;NO,TIME OUT?
014060  001371                          BNE     1$              ;NO
014062  012704  000056                  MOV     #TOSNDB,R4      ;YES, SET ERROR CODE
014066  004737  012634                  CALL    LOG             ;LOG IT
014072  000535                          BR      3$              ;EXIT
014074  113775  014540  000030  4$:     MOVB    BRKWD,@XMDB(R5) ;SEND NULL
014102  005037  014544                  CLR     BRKTO           ;RESET TIME OUT
014106  005304                          DEC     R4              ;MORE NULLS TO SEND?
014110  001355                          BNE     1$              ;YES
014112  005375  000026                  DEC     @XMSR(R5)       ;NO, CLEAR 'BREAK'
014116  017500  000024                  MOV     @RCDB(R5),R0    ;HEAVE 'GARBAGE' 1ST BYTE
014122                                  SETPRI  #PRI00          ;SET TO INTERRUPT FO SURE
014122  012700  000000                                                  MOV     #PRI00,R0
014126  104441                                                          TRAP    C$SPRI
014130                                  SETVEC  TUVECT(R5),#RCVINT,#PRI07 ;SET VECTO INFO
014130  012746  000340                                                  MOV     #PRI07,-(SP)
014134  012746  014450                                                  MOV     #RCVINT,-(SP)
014140  016546  000204                                                  MOV     TUVECT(R5),-(SP)
014144  012746  000003                                                  MOV     #3,-(SP)
014150  104437                                                          TRAP    C$SVEC
014152  062706  000010                                                  ADD     #10,SP
014156  062765  000004  000204          ADD     #4,TUVECT(R5)   ;AND INC TO SND VECTOR
014164                                  SETVEC  TUVECT(R5),#SNDINT,#PRI07;AND SET IT
014164  012746  000340                                                  MOV     #PRI07,-(SP)
```

```
     014170  012746  014434                                                       MOV     #SNDINT,-(SP)
     014174  016546  000204                                                       MOV     TUVECT(R5),-(SP)
     014200  012746  000003                                                       MOV     #3,-(SP)
     014204  104437                                                               TRAP    C$SVEC
     014206  062706  000010                                                       ADD     #10,SP
     014212  162765  000004  000204        SUB     #4,TUVECT(R5)     ;RESET VECTOR ADDR.
     014220  005037  014544                CLR     BRKTO             ;RESET TIME OUT
     014224  012704  014542                MOV     #INITWD,R4        ;USE ADDR. FOR SNDBYT
     014230  010437  014546                MOV     R4,BRKPTR         ;AND SAVE FOR 'WAIT''
     014234  052775  000100  000026        BIS     #BIT6,@XMSR(R5)   ;ENABLE INTER.
     014242  004737  014504                CALL    WAIT              ;AND ENTER LOOP
     014246  005715                        TST     @R5               ;ABORTED FROM TIME OUT?
     014250  100446                        BMI     3$                ;YES-EXIT

     014252  005037  014544                CLR     BRKTO             ;RESET TIME OUT
     014256  012704  014542                MOV     #INITWD,R4        ;SEND SECOND INIT
     014262  010437  014546                MOV     R4,BRKPTR         ;SAVE POINTER AGAIN
     014266  052775  000100  000026        BIS     #BIT6,@XMSR(R5)   ;AND THEN ENABLE INT
     014274  004737  014504                CALL    WAIT              ;AND WAIT
     014300  005715                        TST     @R5               ;IF ABORTED
     014302  100431                        BMI     3$                ;THEN EXIT

     014304  012704  014543                MOV     #INITWD+1,R4      ;WHERE RESPONSE WILL GO (ADDRESS)
     014310  010437  014546                MOV     R4,BRKPTR         ;AND FOR 'WAIT'
     014314  052775  000100  000022        BIS     #BIT6,@RCSR(R5)   ;ENABLE RECIEVE INT.
     014322  004737  014504                CALL    WAIT              ;GET ANSWER
     014326  005715                        TST     @R5               ;ABORTED?
     014330  100416                        BMI     3$                ;YES.

     014332  123727  014543  000020        CMPB    INITWD+1,#RSCONT  ;NO, IS IT 'CONTINUE'?
     014340  001003                        BNE     2$                ;NOPE-ERROR

     014342  042715  040000                BIC     #BIT14,@R5        ;SUCCESSFUL, CLEAR DOBREAK FLAG
     014346  000407                        BR      3$                ;EXIT

     014350  113737  014543  003307  2$:   MOVB    INITWD+1,SYSTAT+1         ;SAVE BUM RESPONSE
     014356  012704  000032                MOV     #CNINIT,R4                ;CAN'T INIT CODE
     014362  004737  012634                CALL    LOG               ;LOG IT
                                                                     ;SCHEDULER WILL TRY AGAIN IF NOT ABORTED

     014366  042775  000100  000026  3$:   BIC     #BIT6,@XMSR(R5)   ;CLEAR INTERRUPTS
     014374  042775  000100  000022        BIC     #BIT6,@RCSR(R5)   ; AND FOR RECIEVE
     014402                                CLRVEC  TUVECT(R5)        ;RELEASE RECIEVE VECT.
     014402  016500  000204                                                       MOV     TUVECT(R5),R0
     014406  104436                                                               TRAP    C$CVEC
     014410  062765  000004  000204        ADD     #4,TUVECT(R5)     ;AND GET SEND ADDR.
     014416                                CLRVEC  TUVECT(R5)        ;AND RELEASE IT
     014416  016500  000204                                                       MOV     TUVECT(R5),R0
     014422  104436                                                               TRAP    C$CVEC
     014424  162765  000004  000204        SUB     #4,TUVECT(R5)     ;RESTORE POINTER
     014432  000207                        RETURN                    ;RETURN
```

```
                                .SBTTL  INTERRUPT SERVICE ROUTINES AND TIMER

                        BGNSRV  SNDINT                  ;"SEND" INTERRUPT SERVICE:
014434                                                                      SNDINT::
014434
014434  042775  000100  000026  SNDHND: BIC     #BIT6,@XMSR(R5) ;DISABLE INTERRUPT
014442  112475  000030                  MOVB    (R4)+,@XMDB(R5);OUTPUT BYTE
014446                          ENDSRV
014446                                                                      L10004:
014446  000002                                                                      RTI

                        BGNSRV  RCVINT                  ;"RCV" INTERRUPT SERVICE:
014450                                                                      RCVINT::
014450
014450  042775  000100  000022  RCVHND: BIC     #BIT6,@RCSR(R5) ;DISABLE INTS
014456  017565  000024  000074          MOV     @RCDB(R5),DLV(R5)       ;SAVE WORD
014464  116524  000074          MOVB    DLV(R5),(R4)+   ;BYTE TO BUFFER
014470  005765  000074          TST     DLV(R5)         ;ERROR?
014474  100402                  BMI     10$             ;YES
014476  005065  000074          CLR     DLV(R5)         ;NO CLEAR ERROR
014502                  10$:
014502                          ENDSRV
014502                                                                      L10005:
014502  000002                                                                      RTI

014504  000240          WAIT:   NOP                     ;WAIT  LOOP FOR
                                                        ;INTERRUPT SERVICING
014506  020437  014546          CMP     R4,BRKPTR       ;IF=,THEN NO INTERRUPT
014512  001011                  BNE     1$              ;GOT ONE!
014514                          BREAK                   ;SUPERVISOR BREAK
014514  104422                                                      TRAP    C$BRK
014516                          BREAK                   ;KILL SOME TIME
014516  104422                                                      TRAP    C$BRK
014520  005337  014544          DEC     BRKTO           ;TIME OUT?
014524  001367                  BNE     WAIT            ;NO...CONT.
014526  012704  000050          MOV     #TORCVB,R4      ;YES LOAD ERROR #
014532  004737  012634          CALL    LOG             ;LOG IT
014536  000207          1$:     RETURN                  ;RETURN

014540  000000          BRKWD:  .WORD   0               ;NULL
014542     004          INITWD: .BYTE   RSINIT          ;INIT COMMAND
014543     000                  .BYTE   0               ;RSCONT IS EXPECTED HERE
014544  000000          BRKTO:  .WORD   0               ;TIME OUT
014546  000000          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:  - 7(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
                                ;--

014550                          COMPAR:: PUSH   R0              ;COMPARE DATA IS DATA PACKET
014550  010046                                                  MOV     R0,-(SP)

014552                                   PUSH   R4              ;TO PATTERN WRITTEN
014552  010446                                                  MOV     R4,-(SP)

014554                                   PUSH   R1              ;USING BYTE COUNT IN PACKET
014554  010146                                                  MOV     R1,-(SP)

014556  005037  014726                  CLR    BDBYTS          ;CLEAR TOTAL WRONG
014562  016504  000104                  MOV    PKPTR(R5),R4    ;GET TOP OF PACKET
014566  005737  002216                  TST    CMPDAT          ;COMPARE SELECTED?
014572  001451                          BEQ    4$              ;NO-EXIT
014574  005204                          INC    R4              ;YES, LOCATE COUNT
014576  111401                          MOVB   @R4,R1          ;GET IT
014600  042701  177400                  BIC    #177400,R1      ;SIGN-UNEXTEND
                                                               ;MUST TEST BYTE-WISE...
014604  005204                          INC    R4              ;-->FIRST DATA  BYTE
014606  126524  000072          1$:     CMPB   PATTEN(R5),(R4)+  ;DATA-WHAT WAS EXPECTED?
014612  001402                          BEQ    2$              ;YES
014614  005237  014726                  INC    BDBYTS          ;NO, INCREMENT TOTAL WRONG
014620  005301                  2$:     DEC    R1              ;MORE LEFT?
014622  001371                          BNE    1$              ;YES
014624  005737  014726                  TST    BDBYTS          ;ANY WRONG?
014630  001432                          BEQ    4$              ;NO
014632  011537  002074                  MOV    @R5,L$LUN       ;GET UNIT NUMBER
014636  042737  177770  002074          BIC    #177770,L$LUN   ;MASK IT OFF
014644                                  ERRSOFT 0.,MSBDA,ERRDES  ;YES-PRINT 'BAD DATA IN PACKET' ERROR
014644  104457                                                          TRAP    C$ERSOFT
014646  000000                                                          .WORD   0
014650  002340                                                          .WORD   MSBDA
014652  013170                                                          .WORD   ERRDES
014654                                  PRINTB  #DESC,BDBYTS
014654  013746  014726                                                  MOV     BDBYTS,-(SP)
014660  012746  014730                                                  MOV     #DESC,-(SP)
014664  012746  000002                                                  MOV     #2,-(SP)
014670  010600                                                          MOV     SP,R0
014672  104414                                                          TRAP    C$PNTB
014674  062706  000006                                                  ADD     #6,SP
014700  052715  000100                  BIS    #BIT6,@R5       ;LET 'EM KNOW UPSTAIRS-BAD DATA FLAG
```

```
3054 014704  012737  000204  003336          MOV     #132.,PRNSIZ    ;SIZE IS ONE DATA PACK
3055 014712  004737  014764                  CALL    PRNPAK          ;AND PRINT THE PACKET
3056 014716                          4$:     POP     R1              ;RESTORE
     014716  012601                                                  MOV     (SP)+,R1

3057 014720                                  POP     R4
     014720  012604                                                  MOV     (SP)+,R4

3058 014722                                  POP     R0
     014722  012600                                                  MOV     (SP)+,R0

3059
3060 014724  000207                          RETURN
3061
3062 014726  000000                  BDBYTS: .WORD
3063 014730     045      101      124 DESC:   .ASCIZ  /%ATOTAL BAD BYTES= %D3%A.%N/
3064                                         .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
                                ;--

014764  000240                  PRNPAK:: NOP                    ;PRINTS 1 PACKET
                                                                ;PKPTR(R5)->TOP OF PACKET
                                                                ;PRNSIZ (PASSED)=BYTE COUNT
014766                                  PUSH    R0
014766  010046                                                  MOV     R0,-(SP)

014770                                  PUSH    R4
014770  010446                                                  MOV     R4,-(SP)

014772  105737  002214                  TSTB    PRBUF           ;PRINT PACKET SELECTED?
014776  001451                          BEQ     4$              ;NO
015000  016504  000104                  MOV     PKPTR(R5),R4    ;YES-GET TOP OF PACK
015004  012737  000020  015130  1$:     MOV     #16.,LNCNT      ;16 BYTES PER LINE
015012  112437  015132          2$:     MOVB    (R4)+,PRDAT     ;AVOID SIGN EXTEND
015016                                  PRINTF  #PRFORM,<B,PRDAT>  ;PRINT BYTE
015016  005046                                                          CLR     -(SP)
015020  153716  015132                                                  BISB    PRDAT,(SP)
015024  012746  015134                                                  MOV     #PRFORM,-(SP)
015030  012746  000002                                                  MOV     #2,-(SP)
015034  010600                                                          MOV     SP,R0
015036  104417                                                          TRAP    C$PNTF
015040  062706  000006                                                  ADD     #6,SP
015044  005337  003336                  DEC     PRNSIZ          ;ONE LESS
015050  001414                          BEQ     3$              ;NO MORE
015052  005337  015130                  DEC     LNCNT           ;NEW LINE?
015056  001355                          BNE     2$              ;NOT YET
015060                                  PRINTF  #CARLF          ;YES
015060  012746  015144                                                  MOV     #CARLF,-(SP)
015064  012746  000001                                                  MOV     #1,-(SP)
015070  010600                                                          MOV     SP,R0
015072  104417                                                          TRAP    C$PNTF
015074  062706  000004                                                  ADD     #4,SP
015100  000741                          BR      1$              ;NEXT LINE
015102                          3$:     PRINTF  #CARLF          ;FINISH UP
015102  012746  015144                                                  MOV     #CARLF,-(SP)
015106  012746  000001                                                  MOV     #1,-(SP)
015112  010600                                                          MOV     SP,R0
015114  104417                                                          TRAP    C$PNTF
015116  062706  000004                                                  ADD     #4,SP
015122                          4$:     POP     R4
015122  012604                                  MOV     (SP)+,R4

015124                                  POP     R0
015124  012600                                  MOV     (SP)+,R0

015126  000207                          RETURN                  ;RETURN
```

```
3097
3098 015130  000000                        LNCNT:  .WORD
3099 015132  000000                        PRDAT:  .WORD
3100 015134     045      117      063      PRFORM: .ASCIZ  /%03%A /
3101                                               .EVEN
3102 015144     045      116      000      CARLF:  .ASCIZ  /%N/
3103                                               .EVEN
3104
3105 015150                                        ENDMOD
3106
```

```
                                .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$BRK
                                        MOV     #BLKTBL,RPTR    ;GET 1ST DEVICE BLOCK
                                        PRINTS  #STATHD         ;HEADER
                                                                                MOV     #STATHD,-(SP)
                                                                                MOV     #1,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTS
                                                                                ADD     #4,SP
                                        BREAK                   ;^C CHECK
                                                                                TRAP    C$BRK
                                        PRINTS  #STHD2          ;2ND HEADER
                                                                                MOV     #STHD2,-(SP)
                                                                                MOV     #1,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTS
                                                                                ADD     #4,SP
                                1$:     BREAK                   ;^C CHECK
                                                                                TRAP    C$BRK
                                        MOV     @RPTR,R5        ;GET DEVICE BLOCK
```

```
                                                BIT     #BIT11,@R5      ;UNIT NOT TESTED?
                                                BNE     2$              ;TRUE, DON'T PRINT STATISTICS
                                                                        ;OK TO PRINT
                                                MOV     @R5,RLUN        ;SAVE STATUS WORD
                                                BIC     #177770,RLUN    ;MASK UNIT NUM.
                                                MOVB    SOFTR(R5),R1    ;SOFTREAD
                                                BIC     #177400,R1      ;SIGN-UNEXTEND
                                                MOVB    SOFTW(R5),R2    ;SOFT WRITE
                                                BIC     #177400,R2      ;
                                                MOVB    HARDR(R5),R3    ;HARD READ
                                                BIC     #177400,R3      ;
                                                MOVB    HARDW(R5),R4    ;HARD WRITE
                                                BIC     #177400,R4      ;
                                                PRINTS  #FM0,RLUN       ;SUMMARY/UNIT #
                                                                                MOV     RLUN,-(SP)
                                                                                MOV     #FM0,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTS
                                                                                ADD     #6,SP
                                                PRINTS  #FM,#0,WRTN0(R5),RDN0(R5),<B,BDATA(R5)>,R1,R2,R3,R4
                                                                                MOV     R4,-(SP)
                                                                                MOV     R3,-(SP)
                                                                                MOV     R2,-(SP)
                                                                                MOV     R1,-(SP)
                                                                                CLR     -(SP)
                                                                                BISB    BDATA(R5),(SP)
                                                                                MOV     RDN0(R5),-(SP)
                                                                                MOV     WRTN0(R5),-(SP)
                                                                                MOV     #0,-(SP)
                                                                                MOV     #FM,-(SP)
                                                                                MOV     #11,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTS
                                                                                ADD     #24,SP
                                                MOVB    SOFTR+1(R5),R1  ;SAME
                                                BIC     #177400,R1      ;AS
                                                MOVB    SOFTW+1(R5),R2  ;ABOVE
                                                BIC     #177400,R2      ;THIS
                                                MOVB    HARDR+1(R5),R3  ;TIME
                                                BIC     #177400,R3      ;FOR
                                                MOVB    HARDW+1(R5),R4  ;DRIVE
                                                BIC     #177400,R4      ;ONE

                                                PRINTS  #FM,#1,WRTN1(R5),RDN1(R5),<B,BDATA+1(R5)>,R1,R2,R3,R4
                                                                                MOV     R4,-(SP)
                                                                                MOV     R3,-(SP)
                                                                                MOV     R2,-(SP)
                                                                                MOV     R1,-(SP)
                                                                                CLR     -(SP)
                                                                                BISB    BDATA+1(R5),(SP)
                                                                                MOV     RDN1(R5),-(SP)
                                                                                MOV     WRTN1(R5),-(SP)
                                                                                MOV     #1,-(SP)
                                                                                MOV     #FM,-(SP)
                                                                                MOV     #11,-(SP)
                                                                                MOV     SP,R0
```

```
                015526  104416                                                                          TRAP    C$PNTS
                015530  062706  000024                                                                  ADD     #24,SP
3196            015534  023727  015576  003366  2$:     CMP     RPTR,#LSTDEV    ;ALL UNITS DONE?
3197            015542  103005                          BHIS    3$              ;YES
3198            015544  062737  000002  015576          ADD     #2,RPTR         ;NO-DO
3199
3200            015552  000137  015236                  JMP     1$              ;MORE UNITS
3201
3202            015556                          3$:     POP     R5
                015556  012605                                                  MOV     (SP)+,R5
3203            015560                                  POP     R4
                015560  012604                                                  MOV     (SP)+,R4
3204            015562                                  POP     R3
                015562  012603                                                  MOV     (SP)+,R3
3205            015564                                  POP     R2
                015564  012602                                                  MOV     (SP)+,R2
3206            015566                                  POP     R1
                015566  012601                                                  MOV     (SP)+,R1
3207            015570                                  POP     R0
                015570  012600                                                  MOV     (SP)+,R0
3208            015572                                  ENDRPT
                015572                                                                  L10006:
                015572  104425                                                                  TRAP    C$RPT
3209            015574  000000                  RLUN:   .WORD
3210            015576  000000                  RPTR:   .WORD
3211
3212            015600      045     116     045 STATHD: .ASCII  /%N%A       DR BLKS WR  BLKS RD  BDPAK  /
3213            015646      104     103     110         .ASCIZ  @DCHK/RD DCHK/WR  DCHK/RD DCHK/WR%N@
3214                                                    .EVEN
3215            015712      045     101     125 FM0:    .ASCIZ  /%AUNIT %D1%N/
3216                                                    .EVEN
3217
3218            015730      045     101     040 FM:     .ASCII  /%A       %D1%A %D5%A.    %D5%A.     %D3%A.     /
3219            016004      045     104     063         .ASCIZ  /%D3%A.   %D3%A.      %D3%A.     %D3%A.%N/
3220                                                    .EVEN
3221            016054      045     101     040 STHD2:  .ASCII  /%A                                          /
3222            016121      122     105     103         .ASCIZ  /RECOV   RECOV   UNRECOV UNRECOV%N/
3223                                                    .EVEN
3224            016164                                  ENDMOD
```

```
                                .SBTTL  INITIALIZE SECTION

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

                                        BGNINIT
                                                                                        L$INIT::

                                INIT:   NOP                     ;
                                        CLRB    STRT            ;FOR STATS CLEAR
                                        CLR     TEST9           ;***** CLR TST 9 FLAG
                                        READEF  #EF.START       ;START COMMAND?
                                                                                        MOV     #EF.START,R0
                                                                                        TRAP    C$REFG
                                        BNCOMPLETE INIT2        ;NO
                                                                                        BCC     INIT2
                                        INC     STRT            ;YES, SET START FLAG
                                INIT2:  MOV     #BLKTBL,DEVPTR  ;SET ALL UNITS ABORTED:
                                        CLR     R4              ;UNIT NUMBER
                                1$:     MOV     @DEVPTR,R5      ;GET POINTER
                                        MOV     R4,@R5          ;INSERT UNIT #
                                        BIS     #BIT15,@R5        ;SET ABORTED
                                        BIS     #BIT11,@R5      ;SET UNIT NOT TESTED
                                        ASL     R4              ;*2 FOR LOOK-UP
                                        MOV     BUFTBL(R4),RCVBUF(R5)   ;SETUP POINTER TO UNIT'S BUFFER
                                        ASR     R4              ;CORRECT BACK TO UNIT #
                                        CMP     DEVPTR,#LSTDEV  ;LAST DEVICE DONE?
                                        BHIS    CHECK           ;YES
                                        ADD     #2,DEVPTR       ;NO-GET
                                        INC     R4              ;NEXT DEVICE AND
                                        BR      1$              ;SERVICE

                                CHECK:  CMP     #8.,L$UNIT      ;MAKE SURE NOT
                                        BHIS    GETHRD          ;TOO MANY UNITS
                                        ERRSF   101.,TOMANY             ;TOMANY-REQUEST ^C
                                                                                        TRAP    C$ERSF
                                                                                        .WORD   101
                                                                                        .WORD   TOMANY
                                                                                        .WORD   0
                                        DOCLN                   ;EXIT
                                                                                        TRAP    C$DCLN

                                GETHRD: MOV     #BLKTBL,DEVPTR  ;INIT TABLE POINTER
                                        CLR     R4              ;CLEAR DEVICE COUNTER
                                1$:     MOV     @DEVPTR,R5      ;GET STATUS WORD
                                        MOV     R4,L$LUN        ;UNIT NUM. IN CASE ERROR
                                        GPHARD  R4,R2           ;GET HARD INFO
                                                                                        MOV     R4,R0
                                                                                        TRAP    C$GPHRD
                                                                                        MOV     R0,R2
                                        BNCOMPLETE 3$
                                                                                        BCC     3$
                                        BIC     #BIT11,@R5      ;UNIT IS TESTED!
                                        MOV     (R2)+,R3        ;R3=CSR
                                        MOV     (R2)+,TUVECT(R5)  ;GET VECTOR ADDRESS
```

```
3274 016360  112265  000061                  MOVB    (R2)+,DR+1(R5)  ;SAVE UNIT SUMMARY
3275 016364  005202                          INC     R2              ;GET TO WORD BOUND
3276 016366  012237  016750                  MOV     (R2)+,PDTFLG    ;AND GET PDT FLAG
3277 016372  052715  040000                  BIS     #BIT14,@R5      ;SET SEND BREAK FLAG
3278 016376  032765  000400  000060          BIT     #BIT8,DR(R5)    ;DRIVE 0?
3279 016404  001011                          BNE     13$             ;YES
3280 016406  032765  001000  000060          BIT     #BIT9,DR(R5)    ;DRIVE 1?
3281 016414  001005                          BNE     13$             ;OK
3282 016416                                  ERRSF   102.,NODRVS             ;NEITHER?!
     016416  104454                                                          TRAP    C$ERSF
     016420  000146                                                          .WORD   102
     016422  016714                                                          .WORD   NODRVS
     016424  000000                                                          .WORD   0
3283 016426                                  DOCLN                   ;EXIT
     016426  104444                                                          TRAP    C$DCLN
3284
3285 016430  105737  016746          13$:    TSTB    STRT            ;START COMMAND?
3286 016434  001412                          BEQ     14$             ;NO, DONT CLEAR
3287                                                                 ;YES-CLEAR STATS
3288 016436  012702  000202                  MOV     #BLKEND,R2      ;R2-->END OF STATS
3289 016442  012701  000110                  MOV     #WRTNO,R1       ;FORM ADDRESS OF START·
3290 016446  060501                          ADD     R5,R1           ;R1-->START OF STATS.
3291 016450  162702  000110                  SUB     #WRTNO,R2       ;FORM # TO CLEAR
3292
3293 016454  105021                  2$:     CLRB    (R1)+           ;CLEAR 'EM
3294 016456  005302                          DEC     R2              ;MORE?
3295 016460  001375                          BNE     2$              ;YES
3296 016462  042715  100000          14$:    BIC     #BIT15,@R5        ;SET NOT ABORTED
3297 016466  010365  000022                  MOV     R3,RCSR(R5)     ;GET DEVICE REGISTERS:
3298 016472  062703  000002                  ADD     #2,R3
3299 016476  010365  000024                  MOV     R3,RCDB(R5)
3300 016502  062703  000002                  ADD     #2,R3
3301 016506  010365  000026                  MOV     R3,XMSR(R5)
3302 016512  062703  000002                  ADD     #2,R3
3303 016516  105737  016750                  TSTB    PDTFLG          ;UNIT A PDT?
3304 016522  001402                          BEQ     4$              ;NO
3305 016524  162703  000004                  SUB     #4,R3           ;YES...RCDB=XMDB
3306 016530  010365  000030          4$:     MOV     R3,XMDB(R5)
3307 016534  005065  000072                  CLR     PATTEN(R5)      ;ZERO DATA PATTERN
3308 016540  005065  000002                  CLR     RETRY(R5)       ;NO RETRIES
3309 016544  005065  000064                  CLR     REC(R5)         ;NO RECORD
3310 016550  005065  000076                  CLR     SUCCS(R5)       ;NO SUCCESS
3311 016554  005065  000074                  CLR     DLV(R5)         ;NO DLV ERROR
3312 016560  005065  000210                  CLR     MRSP(R5)        ;***** CLR MRSP INDICATOR
3313 016564  005037  003340                  CLR     ALLGON          ;OK TO PRINT STATISTICS
3314 016570  062737  000002  003312  3$:     ADD     #2,DEVPTR       ;-->NEXT DEVICE
3315 016576  005204                          INC     R4              ;INCREMENT UNIT NUMBER
3316 016600  020437  002012                  CMP     R4,L$UNIT       ;MORE UNITS?
3317 016604  001250                          BNE     1$              ;YES, GP HARD THE NEXT
3318
3319 016606  005037  003306                  CLR     SYSTAT          ;SYSTEM STATUS WORD
3320 016612                                  RFLAGS  FLGLOC          ;GET USER FLAGS
     016612  104421                                                          TRAP    C$RFLA
     016614  010037  016752                                                  MOV     R0,FLGLOC
3321 016620  005037  003332          5$:     CLR     BLKER           ;NO ERROR
3322 016624  013737  002210  003310  SETLEN: MOV     LENGTH,TAPLEN   ;GET # OF RECORDS
3323 016632  006237  003310                  ASR     TAPLEN          ;GET # BLOCKS PER TRACK
```

```
3324 016636  012737  000200  003334          MOV     #200,SECREC     ;PRESET SECOND START AT 200
3325 016644  022737  000200  003310          CMP     #200,TAPLEN     ;# BLKS > 128.?
3326 016652  101003                          BHI     3$              ;NO-SWITCH TRACKS 2ND PASS
3327 016654  012737  000400  003334          MOV     #400,SECREC     ;YES-START AT 400
3337
3349
3350 016662                          3$:     ENDINIT
     016662                                                          L10007:
     016662  104411                                                          TRAP    C$INIT
3351
3352
3353 016664     124     117     117  TOMANY: .ASCIZ  /TOO MANY UNITS MAX.=8 /
3354                                          .EVEN
3355 016714     123     105     114  NODRVS: .ASCIZ  /SELECT AT LEAST 1 DRIVE /
3356                                          .EVEN
3357 016746  000000                  STRT::  .WORD
3358 016750  000000                  PDTFLG::.WORD           ;TU58 IS IN PDT
3359 016752  000000                  FLGLOC::.WORD           ;USER FLAGS
```

```
                                        ;++
                                        ; THE AUTO DROP CODE IS INVOKED WHEN THE ADR FLAG IS SET AND CHECKS FOR
                                        ; A VALID INTERFACE LOCATION.  DROPS UNIT IF INTERFACE IS NOT THERE.
                                        ;--

                                        BGNAUTO
016754                                                                                  L$AUTO::
016754
016754  000240                                NOP             ;AUTO DROP ROUTINE
016756                                      SETVEC  #4,#TRPHND,#PRI07       ;GET BUS TRAP VEC.
016756  012746  000340                                                          MOV     #PRI07,-(SP)
016762  012746  017064                                                          MOV     #TRPHND,-(SP)
016766  012746  000004                                                          MOV     #4,-(SP)
016772  012746  000003                                                          MOV     #3,-(SP)
016776  104437                                                                  TRAP    C$SVEC
017000  062706  000010                                                          ADD     #10,SP
017004  012737  003350  017062              MOV     #BLKTBL,TRPPTR  ;GET TOP OF DATA BLOCK TABLE
017012  017705  000044          1$:         MOV     @TRPPTR,R5      ;GET DATA BLOCK
017016  032715  104000                      BIT     #BIT15!BIT11,@R5  ;NOT TESTED OR ABORTED?
017022  100403                              BMI     2$              ;YES
017024  005775  000022                      TST     @RCSR(R5)       ;NO-VALID ADDRESS?
017030  000240                              NOP                     ;YES...(TRAP IF NOT)
017032  023727  017062  003366    2$:       CMP     TRPPTR,#LSTDEV  ;MORE TO TRY?
017040  103004                              BHIS    3$              ;NO
017042  062737  000002  017062              ADD     #2,TRPPTR       ;ON TO NEXT
017050  000760                              BR      1$              ;GET IT
017052                          3$:         CLRVEC  #4              ;RESTORE
017052  012700  000004                                                          MOV     #4,R0
017056  104436                                                                  TRAP    C$CVEC

                                        ENDAUTO
017060                                                                  L10010:
017060                                                                          TRAP    C$AUTO
017060  104461
017062  000000                          TRPPTR: .WORD


                                        ;ILLEGAL ADDRESS TRAP HANDLER:

017064                                  TRPHND: PRINTF  #MSAUTO         ;SAY ''AUTO DROPPED''
017064  012746  017116                                                          MOV     #MSAUTO,-(SP)
017070  012746  000001                                                          MOV     #1,-(SP)
017074  010600                                                                  MOV     SP,R0
017076  104417                                                                  TRAP    C$PNTF
017100  062706  000004                                                          ADD     #4,SP
017104  011500                              MOV     @R5,R0          ;GET UNIT #
017106  042700  177770                      BIC     #177770,R0      ;MASK IT OFF
017112                                      DODU    R0              ;DROP HIM
017112  104451                                                                  TRAP    C$DODU
017114  000002                              RTI
017116     045     101     101  MSAUTO:     .ASCIZ  /%AAUTO DROP: %N/
```

```
                                .SBTTL  CLEANUP CODING SECTION

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

                                        BGNCLN
                                                                                L$CLEAN::
017136  005737  003340                  TST     ALLGON          ;ENTRANCE FROM ALL-UNITS-ABORTED?
017142  001004                          BNE     1$              ;YES-EXIT
017144  005737  002212                  TST     STAEOP          ;NO-STATS AT EOP?
017150  001401                          BEQ     1$              ;NO
017152                                  DORPT                   ;YES
017152  104424                                                          TRAP    C$DRPT

017154                          1$:     ENDCLN
017154                                                          L10011:
017154  104412                                                          TRAP    C$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
017156                                  PUSH    R0                      ;SAVE IT
017156  010046                                                          MOV     R0,-(SP)

                                        PUSH    R5                      ;SAVE PRESENT UNIT POINTER
017160  010546                                                          MOV     R5,-(SP)

017162  004737  017222                  CALL    GETR5                   ;GET POINTER TO UNIT
017166  052715  100000                  BIS     #BIT15,@R5              ;SET ABORTED
017172                                  POP     R5                      ;RESTORE PRESENT UNIT POINTER
017172  012605                                                          MOV     (SP)+,R5

017174                                  POP     R0                      ;RETRIEVE UNIT NUMBER
017174  012600                                                          MOV     (SP)+,R0

017176                                  PRINTF  #ABOMSG,R0
017176  010046                                                                  MOV     R0,-(SP)
017200  012746  017254                                                          MOV     #ABOMSG,-(SP)
017204  012746  000002                                                          MOV     #2,-(SP)
017210  010600                                                                  MOV     SP,R0
017212  104417                                                                  TRAP    C$PNTF
017214  062706  000006                                                          ADD     #6,SP

                                        ENDDU                                           L10012:
017220                                                                                  TRAP    C$DU
017220  104453
017222  012737  003350  017252  GETR5:   MOV     #BLKTBL,PTR             ;-->UNIT 0
017230  017705  000016          1$:     MOV     @PTR,R5                 ;GET STATUS WORD
017234  005300                          DEC     R0                      ;CORRECT UNIT?
017236  100404                          BMI     2$                      ;YES
017240  062737  000002  017252          ADD     #2,PTR                  ;NO,-->NEXT
017246  000770                          BR      1$                      ;CONTINUE
017250  000207                  2$:     RETURN
017252  000000                  PTR:    .WORD

017254      045     101     104 ABOMSG: .ASCIZ   /%ADROPPED UNIT %D1%N/
                                        .EVEN
```

```
                                .SBTTL  ADD UNIT SECTION

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

017302                                  BGNAU
017302                                                                  L$AU::

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

017302                                  ENDAU
017302                                                                  L10013:
017302  104452                                                                  TRAP    C$AU
```

```
                                .SBTTL  TEST 1 / DEVICE SELF-DIAGNOSTIC EXECUTION

017304                                  BGNMOD
                                .NLIST  ME,BEX

                                        BGNTST
                                                                                T1::
017304                                  TSTID   #TST1
017304  012737  017350  003326                        MOV     #TST1,TSTTOP    ;SAVE ADDR OF TEST
017312  004737  006004                                CALL    SETUP           ;INIT UNITS TSTPC
017316  004737  005632                                CALL    SETDR           ;GET 1ST DRVS.
017322  004737  006052                                CALL    RUN             ;DO TEST
017326  004737  005530                                CALL    SWAPDR          ;GET NEXT DRVS.
017332  103004                                        BCC     64$             ;BR NO 2ND DRVS
017334  004737  006004                                CALL    SETUP           ;REINIT UNITS TSTPC
017340  004737  006052                                CALL    RUN             ;REPEAT TEST
017344                                  64$:                                  ;DONE
017344                                  EXIT TST
017344  104432                                                          TRAP    C$EXIT
017346  000136                                                          .WORD   L10014-.

017350                                  TST1:   TUSELF

017350  012700  027722                  64$:    MOV     #TRBUF,R0       ;FORM COMMAND PACKET
017354  112710  000002                          MOVB    #RSCMND,@R0     ;COMMAND FLAG
017360  112760  000012  000001                  MOVB    #RSMSIZ,1(R0)   ;SIZE OF MESSAGE
017366  112760  000007  000002                  MOVB    #RSSSLF,2(R0)   ;SELF TEST OPERATION
017374  105060  000003                          CLRB    3(R0)           ;NO MODIFIER.
017400  005060  000004                          CLR     4(R0)           ;NO DRIVE OR SWITCHES
017404  005060  000006                          CLR     6(R0)           ;NO SEQUENCE NUMBER
017410  005060  000010                          CLR     8.(R0)          ;NO BYTES
017414  005060  000012                          CLR     10.(R0)         ;NO RECORD #
017420  012701  000012                          MOV     #RSMSIZ,R1      ;GET SIZE
017424  005721                                  TST     (R1)+           ;+2 FOR CHECKSUM
017426  012765  000016  000070                  MOV     #RSSNSZ,SNDCNT(R5)  ;SIZE TO SEND
017434  004737  013654                          CALL    CHKSUM          ;FORM CHECKSUM
017440  010110                                  MOV     R1,(R0)         ;INSERT INTO PACKET
017442  012765  000002  000034                  MOV     #RSEND,XSFLG(R5)  ;EXPECT END.
017450  012765  000016  000036                  MOV     #RSNDSZ,XSCNT(R5)  ;THIS BIG
017456  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
                                                                        ;SEND
017464  004737  006560                          CALL    RSVP            ;RETURN TO SCHEDULER
017470  032715  000010                          BIT     #BIT3,@R5       ;RETRY?(BAD FLAG)
017474  001325                                  BNE     64$             ;YES

017476  005237  003322          INC     DONE
017502  000207                  RETURN

                                        ENDTST
017504                                                          L10014:
017504                                                                  TRAP    C$ETST
017504  104401
```

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

                                        BGNTST
                                                                                T2::
                                        TSTID   #TST2
017506 012737 017552 003326                             MOV     #TST2,TSTTOP    ;SAVE ADDR OF TEST
017514 004737 006004                                    CALL    SETUP           ;INIT UNITS TSTPC
017520 004737 005632                                    CALL    SETDR           ;GET 1ST DRVS.
017524 004737 006052                                    CALL    RUN             ;DO TEST
017530 004737 005530                                    CALL    SWAPDR          ;GET NEXT DRVS.
017534 103004                                           BCC     64$             ;BR NO 2ND DRVS
017536 004737 006004                                    CALL    SETUP           ;REINIT UNITS TSTPC
017542 004737 006052                                    CALL    RUN             ;REPEAT TEST
017546                                          64$:                            ;DONE
                                        EXIT TST
017546 104432                                                           TRAP    C$EXIT
017550 000206                                                           .WORD   L10015-.

017552 005004                   TST2:   CLR     R4                      ;R4=INDEX INTO RECORD TABLE
017554 016465 017742 000064     1$:     MOV     RECDAT(R4),REC(R5)      ;GET THE RECORD

017562                                  TUSEEK  REC(R5),DR(R5)  ;SEEK IT

017562 012700 027722                            64$:    MOV     #TRBUF,R0       ;-->(POINT TO) XMIT BUFFER
017566 112710 000002                                    MOVB    #RSCMND,@R0     ;FORM COMMAND MESSAGE PACK
017572 112760 000012 000001                             MOVB    #RSMSIZ,1(R0)   ;THIS BIG
017600 112760 000005 000002                             MOVB    #RSSSEK,2(R0)   ;OP CODE IS SEEK
017606 016560 000064 000012                             MOV     REC(R5),10.(R0) ;TO THIS RECORD
017614 116560 00006C 000004                             MOVB    DR(R5),4.(R0)   ;AND WHICH DRIVE
017622 105060 000003                                    CLRB    3.(R0)          ;NO MODIFIER
017626 105060 000005                                    CLRB    5.(R0)          ;NO SWITCHES
017632 005060 000006                                    CLR     6.(R0)          ;NO SEQUENCE #
017636 005060 000010                                    CLR     8.(R0)          ;NO BYTE COUNT
017642 012701 000012                                    MOV     #RSMSIZ,R1      ;GET COUNT
017646 005721                                           TST     (R1)+           ;PLUS FLAG + BCNT
                                                                                ;FOR CHECKSUM CALC
017650 004737 013654                                    CALL    CHKSUM          ;R0-->TOP   R1=# OF BYTES
017654 010110                                           MOV     R1,(R0)         ;INSERT INTO PACKET
                                                                                ;SET UP EXPECTATIONS:
017656 012765 000016 000070                             MOV     #RSSNSZ,SNDCNT(R5)  ;HOW MANY TO SEND
017664 112765 000002 000034                             MOVB    #RSCMND,XSFLG(R5)   ;EXPECT END PACK
017672 012765 000016 000036                             MOV     #RSNDSZ,XSCNT(R5)   ;COUNT WITH THIS
017700 012765 000001 000032                             MOV     #1.,XSPKNM(R5)  ;EXPECT ONLY 1 PACKET

017706 004737 006560                                    CALL    RSVP            ;SEND
                                                                                ;AND RETURN TO SCHEDULER
017712 032715 000010                                    BIT     #BIT3,@R5       ;RETRY (FLAG BYTE ERROR)?
017716 001321                                           BNE     64$             ;YES

017720 062704 000002            ADD     #2,R4           ;POINT TO NEXT RECORD
017724 026427 017742 177777     CMP     RECDAT(R4),#-1. ;LAST ONE DONE?
017732 001310                   BNE     1$              ;NO-LOOP
017734 005237 003322            INC     DONE            ;YES-SET DONE FLAG
017740 000207                   RETURN
```

```
 3608 017742  000000                RECDAT: 0.          ;BOT
 3609 017744  000200                        200         ;BOT OTHER TRACK
 3610 017746  000177                        177         ;EOT
 3611 017750  000377                        377         ;EOT OTHER TRACK
 3612 017752  000400                        400         ;BOT AGAIN
 3613 017754  177777                        -1.
 3614 017756                                ENDTST
      017756                                                L10015:
      017756  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
017760                                                          MOV     #TST3,TSTTOP    ;SAVE ADDR OF TEST
017766                                                          CALL    SETUP           ;INIT UNITS TSTPC
017772                                                          CALL    SETDR           ;GET 1ST DRVS.
017776                                                          CALL    RUN             ;DO TEST
020002                                                          CALL    SWAPDR          ;GET NEXT DRVS.
020006                                                          BCC     64$             ;BR NO 2ND DRVS
020010                                                          CALL    SETUP           ;REINIT UNITS TSTPC
020014                                                          CALL    RUN             ;REPEAT TEST
020020                                                  64$:                    ;DONE
                                        EXIT TST
                                                                                        TRAP    C$EXIT
020022                                                                                  .WORD   L10016-.

020024  012765  000100  000066  TST3:   MOV     #100,TMP(R5)    ;INIT TO HALF OF REMAINING
020032  005004                          CLR     R4              ;FOR INDEX INTO DATA TABLE
020034  005065  000064                  CLR     REC(R5)         ;START AT RECORD 0
020040  016465  022742  000072  1$:     MOV     TST3PT(R4),PATTEN(R5)   ;GET DATA
020046                                  TUWRIT  PATTEN(R5),REC(R5),#512.,DR(R5),#0
                                                        72$:    MOV     #TRBUF,R0       ;MAKE COMMAND PACKET:
020046  012700  027722                                          MOVB    #RSCMND,@R0     ;COMMAND FLAG
020052  112710  000002                                          MOVB    #RSMSIZ,1(R0)   ;THIS SIZE
020056  112760  000012  000001                                  MOVB    #RSSWR,2(R0)    ;INSERT OP CODE-WRITE
020064  112760  000003  000002                                  MOVB    #0,3.(R0)       ;VERIFY (1 OR 0)
020072  112760  000000  000003                                  MOVB    DR(R5),4.(R0)   ;DRIVE #
020100  116560  000060  000004                                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE SWITCH
020106  112760  000020  000005                                  CLR     6.(R0)          ;NO SEQUENCE #
020114  005060  000006                                          MOV     #512.,8.(R0)    ;TOTAL COUNT TO WRITE
020120  012760  001000  000010                                  MOV     REC(R5),10.(R0) ;AT  RECORD N
020126  016560  000064  000012                                  MOV     #RSMSIZ,R1      ;THE PACKET SIZE PLUS+2
020134  012701  000012                                          TST     (R1)+           ;(FLAG AND COUNT) INTO R1
020140  005721                                                  MOV     #RSSNSZ,SNDCNT(R5)  ;LOAD THE  SIZE TO SEND
020142  012765  000016  000070                                  CALL    CHKSUM          ;R0 --> R1=COUNT
020150  004737  013654                                          MOV     R1,(R0)         ;PUT CHKSUM IN PACKET
020154  010110                                                                  ;SET UP EXPECTATIONS:

020156  012765  000020  000034                                  MOV     #RSCONT,XSFLG(R5)   ;THE FLAG
020164  012765  000001  000036                                  MOV     #1,XSCNT(R5)    ;THE COUNT
020172  012765  000001  000052                                  MOV     #1,XSPKNM(R5)   ;THE # PACKETS EXPECTED
020200  012702  001000                                          MOV     #512.,R2                ;GET # OF DATA BYTES
020204  004737  006560                                          CALL    RSVP            ;SEND (AND RETURN TO SCHEDULER)
020210  032715  000010                                          BIT     #BIT3,@R5       ;FLAG BYTE ERROR?
020214  001314                                                  BNE     72$             ;YES
020216  042715  010000                                          BIC     #BIT12,@R5      ;FLAG FOR LAST PACKET
020222  012700  027722                                  64$:    MOV     #TRBUF,R0       ;POINT TO TOP OF BUFFER AGAIN
020226  020227  000200                                          CMP     R2,#128.        ;START DATA PACKET(S)
020232  101004                                                  BHI     65$             ;#512. > 128.!
020234  010201                                                  MOV     R2,R1           ;#512.<128.
020236  052715  010000                                          BIS     #BIT12,@R5      ;SO LAST PACKET NOW
020242  000402                                                  BR      66$             ;USE REMAINING COUNT
```

```
020244  012701  000200                  65$:    MOV     #128.,R1        ;USE 128. BYTES
020250  110160  000001                  66$:    MOVB    R1,1(R0)        ;COPY COUNT TO BUFFER
020254  010103                                  MOV     R1,R3           ;R3=COUNTER TO LOAD BUFFER
020256  112710  000001                          MOVB    #RSDATA,@R0     ;FLAG FIRST
020262  005720                                  TST     (R0)+           ;SKIP COUNT
020264  116520  000072                  67$:    MOVB    PATTEN(R5),(R0)+        ;INSERT DATA
020270  005303                                  DEC     R3              ;MORE?
020272  101374                                  BHI     67$             ;YES
020274  012700  027722                          MOV     #TRBUF,R0       ;-->TOP AGAIN
020300  116001  000001                          MOVB    1(R0),R1        ;GET COUNT
020304  042701  177400                          BIC     #177400,R1      ;ZERO SIGN EXTEND
020310  010165  000070                          MOV     R1,SNDCNT(R5)   ;HOW MANY TO SEND PLUS
020314  062765  000004  000070                  ADD     #4,SNDCNT(R5)   ;FLAG,COUNT,CHKSUM
020322  062701  000002                          ADD     #2,R1           ;COMPENSATE FOR FLAG + COUNT
020326  004737  013654                          CALL    CHKSUM          ;FOR CHECKSUM CALC.
020332  110120                                  MOVB    R1,(R0)+        ;CHKSUM INTO PACKET
020334  000301                                  SWAB    R1              ;EVEN ON AN ODD
020336  110120                                  MOVB    R1,(R0)+        ;BYTE BOUNDARY
020340  032715  010000                          BIT     #BIT12,@R5      ;LAST DATA PACKET?
020344  001412                                  BEQ     68$             ;NO
020346  012765  000002  000034                  MOV     #RSEND,XSFLG(R5)  ;YES-EXPECT 'END'
020354  012765  000016  000036                  MOV     #RSNDSZ,XSCNT(R5)  ;OF THIS SIZE
020362  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
020370  000411                                  BR      69$             ;SEND
020372  012765  000020  000034          68$:    MOV     #RSCONT,XSFLG(R5)  ;(NOT LAST), EXPECT 'CONTINUE'
020400  012765  000001  000036                  MOV     #1,XSCNT(R5)    ;AND 1 BYTE
020406  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
020414  004737  006560                  69$:    CALL    RSVP            ;SEND PACKET
                                                                        ;AND RETURN TO SCHEDULER
020420  032715  000010                          BIT     #BIT3,@R5       ;FLAG BYTE RETRY?
020424  001210                                  BNE     72$             ;YES
020426  032715  002000                          BIT     #BIT10,@R5      ;RETRY DATA ERROR?
020432  001004                                  BNE     70$             ;YES
020434  162702  000200                          SUB     #128.,R2        ;NO, MORE DATA TO SEND?
020440  101270                                  BHI     64$             ;YES
020442  000502                                  BR      71$             ;NO
020444                          70$:    TURTRY  REC(R5),#512.,DR(R5)    ;RETRY HERE


020444  012700  027722                  76$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
020450  112710  000002                          MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
020454  112760  000012  000001                  MOVB    #RSMSIZ,1(R0)   ;THIS BIG
020462  112760  000002  000002                  MOVB    #RSSRD,2(R0)    ;OP CODE-READ
020470  016560  000064  000012                  MOV     REC(R5),10.(R0) ;THIS RECORD
020476  116560  000060  000004                  MOVB    DR(R5),4.(R0)   ;THIS DRIVE
020504  105060  000003                          CLRB    3(R0)           ;PRESET NORM THRESHOLD
020510  105715                                  TSTB    @R5             ;REDUCED?
020512  100002                                  BPL     77$             ;NO
020514  105260  000003                          INCB    3(R0)           ;YES-CHANGE THRESHOLD
020520  012760  001000  000010          77$:    MOV     #512.,8.(R0)    ;# BYTES DESIRED
020526  112760  000020  000005                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE
020534  005060  000006                          CLR     6.(R0)          ;NO SEQUENCE #
020540  012701  000012                          MOV     #RSMSIZ,R1      ;SIZE OF PACKET
020544  005721                                  TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
020546  012765  000016  000070                  MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND

020554  004737  013654                          CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
```

```
020560  010110                                          MOV     R1,(R0)         ;INSERT IN PACKET

020562  012701  001000                                  MOV     #512.,R1                ;SET EXPECTATIONS:
                                                                                ;CALC # OF DATA PACKETS TO EXPECT
020566  012703  000034                                  MOV     #XSFLG,R3       ;OFFSET OF FLAG
020572  060503                                          ADD     R5,R3           ;ABS. ADDR. OF XSFLG
020574  005002                                          CLR     R2              ;PRESET
020576  005202                                  73$:    INC     R2              ;# PACKETS EXPECTED
020600  012723  000001                                  MOV     #RSDATA,(R3)+   ;LOAD XSFLG
020604  012723  000204                                  MOV     #132.,(R3)+     ;AND EXPECT COUNT
020610  162701  000200                                  SUB     #128.,R1        ;NEG RESULT LAST TIME
020614  101401                                          BLOS    75$             ;LAST TIME!
020616  000767                                          BR      73$             ;MORE TO DO
020620  005202                                  75$:    INC     R2              ;ADD ONE FOR END PACK
020622  010265  000032                                  MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
020626  012723  000002                                  MOV     #RSEND,(R3)+    ;EXPECT AN END
020632  012713  000016                                  MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES

020636  004737  006560                                  CALL    RSVP            ;SEND
                                                                                ;AND RETURN TO SCHEDULER

3632 020652                     TUREAD  REC(R5),#512.,DR(R5),#0

020652  012700  027722                          82$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
020656  112710  000002                                  MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
020662  112760  000012  000001                          MOVB    #RSMSIZ,1(R0)   ;THIS BIG
020670  112760  000002  000002                          MOVB    #RSSRD,2(R0)    ;OP CODE IS READ
020676  016560  000064  000012                          MOV     REC(R5),10.(R0) ;THIS RECORD
020704  116560  000060  000004                          MOVB    DR(R5),4.(R0)   ;THIS DRIVE
020712  112760  000000  000003                          MOVB    #0,3.(R0)       ;VERIFY
020720  012760  001000  000010                          MOV     #512.,8.(R0)    ;TOTAL BYTES TO READ
020726  112760  000020  000005                          MOVB    #020,5.(R0)     ;MAINTENANCE MODE
020734  005060  000006                                  CLR     6.(R0)          ;NO SEQUENCE #
020740  012701  000012                                  MOV     #RSMSIZ,R1      ;GET SIZE OF PACKET
020744  005721                                          TST     (R1)+           ;+2 FOR CHECKSUM
020746  012765  000016  000070                          MOV     #RSSNSZ,SNDCNT(R5)  ;SIZE TO SEND
020754  004737  013654                                  CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
020760  010110                                          MOV     R1,(R0)         ;INSERT CHECKSUM

020762  012701  001000                                  MOV     #512.,R1                ;SET EXPECTATIONS:
                                                                                ;CALC # OF DATA PACKETS TO EXPECT:
020766  012703  000034                                  MOV     #XSFLG,R3       ;GET OFFSET
020772  060503                                          ADD     R5,R3           ;ABS. ADDR. OF XSFLG
020774  005002                                          CLR     R2              ;PRESET AS NONE
020776  005202                                  78$:    INC     R2              ;# PACKETS EXPECTED
021000  012723  000001                                  MOV     #RSDATA,(R3)+   ;LOAD XSFLG
021004  012723  000204                                  MOV     #132.,(R3)+     ;AND EXPECTED COUNT
021010  162701  000200                                  SUB     #128.,R1        ;NEG RESULT LAST TIME
021014  101401                                          BLOS    80$             ;LAST TIME
021016  000767                                          BR      78$             ;MORE TO DO
021020  005202                                  80$:    INC     R2              ;ADD ONE FOR END PACK
021022  010265  000032                                  MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
021026  012723  000002                                  MOV     #RSEND,(R3)+    ;EXPECT AN END ALSO...
021032  012713  000016                                  MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES
021036  004737  006560                                  CALL    RSVP            ;SEND
                                                                                ;AND RETURN TO SCHEDULER
```

```
021042  032715  002010                  81$:    BIT     #BIT10!BIT3,@R5 ;RETRY?
021046  001500                                  BEQ     79$             ;NO.
021050                                          TURTRY  REC(R5),#512.,DR(R5)    ;YES


021050  012700  027722                  86$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
021054  112710  000002                          MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
021060  112760  000012  000001                  MOVB    #RSMSIZ,1(R0)   ;THIS BIG
021066  112760  000002  000002                  MOVB    #RSSRD,2(R0)    ;OP CODE-READ
021074  016560  000064  000012                  MOV     REC(R5),10.(R0) ;THIS RECORD
021102  116560  000060  000004                  MOVB    DR(R5),4.(R0)   ;THIS DRIVE
021110  105060  000003                          CLRB    3(R0)           ;PRESET NORM THRESHOLD
021114  105715                                  TSTB    @R5             ;REDUCED?
021116  100002                                  BPL     87$             ;NO
021120  105260  000003                          INCB    3(R0)           ;YES-CHANGE THRESHOLD
021124  012760  001000  000010          87$:    MOV     #512.,8.(R0)    ;# BYTES DESIRED
021132  112760  000020  000005                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE
021140  005060  000006                          CLR     6.(R0)          ;NO SEQUENCE #
021144  012701  000012                          MOV     #RSMSIZ,R1      ;SIZE OF PACKET
021150  005721                                  TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
021152  012765  000016  000070                  MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND

021160  004737  013654                          CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
021164  010110                                  MOV     R1,(R0)         ;INSERT IN PACKET

021166  012701  001000                          MOV     #512.,R1                ;SET EXPECTATIONS:
                                                                        ;CALC # OF DATA PACKETS TO EXPECT
021172  012703  000034                          MOV     #XSFLG,R3       ;OFFSET OF FLAG
021176  060503                                  ADD     R5,R3           ;ABS. ADDR. OF XSFLG
021200  005002                                  CLR     R2              ;PRESET
021202  005202                          83$:    INC     R2              ;# PACKETS EXPECTED
021204  012723  000001                          MOV     #RSDATA,(R3)+   ;LOAD XSFLG
021210  012723  000204                          MOV     #132.,(R3)+     ;AND EXPECT COUNT
021214  162701  000200                          SUB     #128.,R1        ;NEG RESULT LAST TIME
021220  101401                                  BLOS    85$             ;LAST TIME!
021222  000767                                  BR      83$             ;MORE TO DO
021224  005202                          85$:    INC     R2              ;ADD ONE FOR END PACK
021226  010265  000032                          MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
021232  012723  000002                          MOV     #RSEND,(R3)+    ;EXPECT AN END
021236  012713  000016                          MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES

021242  004737  006560                          CALL    RSVP            ;SEND
                                                                        ;AND RETURN TO SCHEDULER

3633 021252  062704  000002                  ADD     #2,R4           ;POINT TO NEXT DATA
3634 021256  005764  022742                  TST     TST3PT(R4)      ;END?
3635 021262  001402                          BEQ     2$              ;YES
3636 021264  000137  020040                  JMP     1$              ;NO-WRITE, READ NEW DATA
3637 021270  005004                  2$:     CLR     R4              ;POINT TO FIRST DATA
3638 021272  062765  000200  000064          ADD     #200,REC(R5)    ;BUT NOW USE ADJACENT RECORD
3639 021300  032765  001000  000064          BIT     #1000,REC(R5)   ;ALL ADJACENT RECORDS DONE?
3640 021306  001002                          BNE     3$              ;YES
3641 021310  000137  020040                  JMP     1$              ;NO-WRITE, READ AT NEW RECORD
3642 021314  162765  001000  000064  3$:     SUB     #1000,REC(R5)   ;RESTORE TO NEXT RECORD
3643 021322  066565  000066  000064          ADD     TMP(R5),REC(R5) ;HALF INTO REST OF TAPE
3644 021330  006265  000066                  ASR     TMP(R5)         ;HALF OF HALF FOR NEXT TIME
3645 021334  103402                          BCS     4$              ;DONE?
```

```
 3646 021336  000137  020040                 JMP     1$              ;NO
 3647 021342  005237  003322          4$:    INC     DONE            ;YES-SET FLAG
 3648 021346  000207                         RETURN
 3649 021350                                 ENDTST
      021350                                                     L10016:
      021350  104401                                                    TRAP    C$ETST
 3650
```

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

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

                                        BGNTST
                                                                                        T4::
                                        TSTID   #TST4
021352  012737  021416  003326                          MOV     #TST4,TSTTOP    ;SAVE ADDR OF TEST
021360  004737  006004                                  CALL    SETUP           ;INIT UNITS TSTPC
021364  004737  005632                                  CALL    SETDR           ;GET 1ST DRVS.
021370  004737  006052                                  CALL    RUN             ;DO TEST
021374  004737  005530                                  CALL    SWAPDR          ;GET NEXT DRVS.
021400  103004                                          BCC     64$             ;BR NO 2ND DRVS
021402  004737  006004                                  CALL    SETUP           ;REINIT UNITS TSTPC
021406  004737  006052                                  CALL    RUN             ;REPEAT TEST
021412                                          64$:                            ;DONE
                                        EXIT TST
021412  104432                                                                  TRAP    C$EXIT
021414  001340                                                                  .WORD   L10017-.

021416  012765  000400  000066  TST4:    MOV     #400,TMP(R5)    ;INIT TO HALF OF REMAINING
021424  005004                          CLR     R4              ;FOR INDEX INTO DATA TABLE
021426  005065  000064                  CLR     REC(R5)         ;START AT RECORD 0
021432  016465  022742  000072  1$:     MOV     TST3PT(R4),PATTEN(R5)   ;GET DATA
                                        TUWRIT  PATTEN(R5),REC(R5),#128.,DR(R5),#BIT7
021440                                          72$:    MOV     #TRBUF,R0       ;MAKE COMMAND PACKET:
021440  012700  027722                          MOVB    #RSCMND,@R0     ;COMMAND FLAG
021444  112710  000002                          MOVB    #RSMSIZ,1(R0)   ;THIS SIZE
021450  112760  000012  000001                  MOVB    #RSSWR,2(R0)    ;INSERT OP CODE-WRITE
021456  112760  000003  000002                  MOVB    #BIT7,3.(R0)    ;VERIFY (1 OR 0)
021464  112760  000200  000003                  MOVB    DR(R5),4.(R0)   ;DRIVE #
021472  116560  000060  000004                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE SWITCH
021500  112760  000020  000005                  CLR     6.(R0)          ;NO SEQUENCE #
021506  005060  000006                          MOV     #128.,8.(R0)    ;TOTAL COUNT TO WRITE
021512  012760  000200  000010                  MOV     REC(R5),10.(R0) ;AT  RECORD N
021520  016560  000064  000012                  MOV     #RSMSIZ,R1      ;THE PACKET SIZE PLUS+2
021526  012701  000012                          TST     (R1)+           ;(FLAG AND COUNT) INTO R1
021532  005721                                  MOV     #RSSNSZ,SNDCNT(R5)  ;LOAD THE  SIZE TO SEND
021534  012765  000016  000070                  CALL    CHKSUM          ;R0 --> R1=COUNT
021542  004737  013654                          MOV     R1,(R0)         ;PUT CHKSUM IN PACKET
021546  010110                                                          ;SET UP EXPECTATIONS:

021550  012765  000020  000034                  MOV     #RSCONT,XSFLG(R5)   ;THE FLAG
021556  012765  000001  000036                  MOV     #1,XSCNT(R5)    ;THE COUNT
021564  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;THE # PACKETS EXPECTED
021572  012702  000200                          MOV     #128.,R2                ;GET # OF DATA BYTES
021576  004737  006560                          CALL    RSVP            ;SEND (AND RETURN TO SCHEDULER)
021602  032715  000010                          BIT     #BIT3,@R5       ;FLAG BYTE ERROR?
021606  001314                                  BNE     72$             ;YES
021610  042715  010000                          BIC     #BIT12,@R5      ;FLAG FOR LAST PACKET
021614  012700  027722                  64$:    MOV     #TRBUF,R0       ;POINT TO TOP OF BUFFER AGAIN
021620  020227  000200                          CMP     R2,#128.        ;START DATA PACKET(S)
021624  101004                                  BHI     65$             ;#128. > 128.!
021626  010201                                  MOV     R2,R1           ;#128.<128.
021630  052715  010000                          BIS     #BIT12,@R5      ;SO LAST PACKET NOW
021634  000402                                  BR      66$             ;USE REMAINING COUNT
```

```
021636  012701  000200                  65$:    MOV     #128.,R1        ;USE 128. BYTES
021642  110160  000001                  66$:    MOVB    R1,1(R0)        ;COPY COUNT TO BUFFER
021646  010103                                  MOV     R1,R3           ;R3=COUNTER TO LOAD BUFFER
021650  112710  000001                          MOVB    #RSDATA,@R0     ;FLAG FIRST
021654  005720                                  TST     (R0)+           ;SKIP COUNT
021656  116520  000072                  67$:    MOVB    PATTEN(R5),(R0)+        ;INSERT DATA
021662  005303                                  DEC     R3              ;MORE?
021664  101374                                  BHI     67$             ;YES
021666  012700  027722                          MOV     #TRBUF,R0       ;-->TOP AGAIN
021672  116001  000001                          MOVB    1(R0),R1        ;GET COUNT
021676  042701  177400                          BIC     #177400,R1      ;ZERO SIGN EXTEND
021702  010165  000070                          MOV     R1,SNDCNT(R5)   ;HOW MANY TO SEND PLUS
021706  062765  000004  000070                  ADD     #4,SNDCNT(R5)   ;FLAG,COUNT,CHKSUM
021714  062701  000002                          ADD     #2,R1           ;COMPENSATE FOR FLAG + COUNT
021720  004737  013654                          CALL    CHKSUM          ;FOR CHECKSUM CALC.
021724  110120                                  MOVB    R1,(R0)+        ;CHKSUM INTO PACKET
021726  000301                                  SWAB    R1              ;EVEN ON AN ODD
021730  110120                                  MOVB    R1,(R0)+        ;BYTE BOUNDARY
021732  032715  010000                          BIT     #BIT12,@R5      ;LAST DATA PACKET?
021736  001412                                  BEQ     68$             ;NO
021740  012765  000002  000034                  MOV     #RSEND,XSFLG(R5)  ;YES-EXPECT 'END'
021746  012765  000016  000036                  MOV     #RSNDSZ,XSCNT(R5)  ;OF THIS SIZE
021754  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
021762  000411                                  BR      69$             ;SEND
021764  012765  000020  000034          68$:    MOV     #RSCONT,XSFLG(R5)  ;(NOT LAST), EXPECT 'CONTINUE'
021772  012765  000001  000036                  MOV     #1,XSCNT(R5)    ;AND 1 BYTE
022000  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
022006  004737  006560                  69$:    CALL    RSVP            ;SEND PACKET
                                                                        ;AND RETURN TO SCHEDULER
022012  032715  000010                          BIT     #BIT3,@R5       ;FLAG BYTE RETRY?
022016  001210                                  BNE     72$             ;YES
022020  032715  002000                          BIT     #BIT10,@R5      ;RETRY DATA ERROR?
022024  001004                                  BNE     70$             ;YES
022026  162702  000200                          SUB     #128.,R2        ;NO, MORE DATA TO SEND?
022032  101270                                  BHI     64$             ;YES
022034  000502                                  BR      71$             ;NO
022036                          70$:    TURTRY  REC(R5),#128.,DR(R5)    ;RETRY HERE


022036  012700  027722                  76$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
022042  112710  000002                          MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
022046  112760  000012  000001                  MOVB    #RSMSIZ,1(R0)   ;THIS BIG
022054  112760  000002  000002                  MOVB    #RSSRD,2(R0)    ;OP CODE-READ
022062  016560  000064  000012                  MOV     REC(R5),10.(R0) ;THIS RECORD
022070  116560  000060  000004                  MOVB    DR(R5),4.(R0)   ;THIS DRIVE
022076  105060  000003                          CLRB    3(R0)           ;PRESET NORM THRESHOLD
022102  105715                                  TSTB    @R5             ;REDUCED?
022104  100002                                  BPL     77$             ;NO
022106  105260  000003                          INCB    3(R0)           ;YES-CHANGE THRESHOLD
022112  012760  000200  000010          77$:    MOV     #128.,8.(R0)    ;# BYTES DESIRED
022120  112760  000020  000005                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE
022126  005060  000006                          CLR     6.(R0)          ;NO SEQUENCE #
022132  012701  000012                          MOV     #RSMSIZ,R1      ;SIZE OF PACKET
022136  005721                                  TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
022140  012765  000016  000070                  MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND

022146  004737  013654                          CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
```

```
022152  010110                                          MOV     R1,(R0)         ;INSERT IN PACKET

022154  012701  000200                                  MOV     #128.,R1                ;SET EXPECTATIONS:
                                                                                ;CALC # OF DATA PACKETS TO EXPECT
022160  012703  000034                                  MOV     #XSFLG,R3       ;OFFSET OF FLAG
022164  060503                                          ADD     R5,R3           ;ABS. ADDR. OF XSFLG
022166  005002                                          CLR     R2              ;PRESET
022170  005202                                  73$:    INC     R2              ;# PACKETS EXPECTED
022172  012723  000001                                  MOV     #RSDATA,(R3)+   ;LOAD XSFLG
022176  012723  000204                                  MOV     #132.,(R3)+     ;AND EXPECT COUNT
022202  162701  000200                                  SUB     #128.,R1        ;NEG RESULT LAST TIME
022206  101401                                          BLOS    75$             ;LAST TIME!
022210  000767                                          BR      73$             ;MORE TO DO
022212  005202                                  75$:    INC     R2              ;ADD ONE FOR END PACK
022214  010265  000032                                  MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
022220  012723  000002                                  MOV     #RSEND,(R3)+    ;EXPECT AN END
022224  012713  000016                                  MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES

022230  004737  006560                                  CALL    RSVP            ;SEND
                                                                                ;AND RETURN TO SCHEDULER

3667 022244                             TUREAD  REC(R5),#128.,DR(R5),#BIT7

022244  012700  027722                          82$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
022250  112710  000002                                  MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
022254  112760  000012  000001                          MOVB    #RSMSIZ,1(R0)   ;THIS BIG
022262  112760  000002  000002                          MOVB    #RSSRD,2(R0)    ;OP CODE IS READ
022270  016560  000064  000012                          MOV     REC(R5),10.(R0) ;THIS RECORD
022276  116560  000060  000004                          MOVB    DR(R5),4.(R0)   ;THIS DRIVE
022304  112760  000200  000003                          MOVB    #BIT7,3.(R0)    ;VERIFY
022312  012760  000200  000010                          MOV     #128.,8.(R0)    ;TOTAL BYTES TO READ
022320  112760  000020  000005                          MOVB    #020,5.(R0)     ;MAINTENANCE MODE
022326  005060  000006                                  CLR     6.(R0)          ;NO SEQUENCE #
022332  012701  000012                                  MOV     #RSMSIZ,R1      ;GET SIZE OF PACKET
022336  005721                                          TST     (R1)+           ;+2 FOR CHECKSUM
022340  012765  000016  000070                          MOV     #RSSNSZ,SNDCNT(R5)  ;SIZE TO SEND
022346  004737  013654                                  CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
022352  010110                                          MOV     R1,(R0)         ;INSERT CHECKSUM

022354  012701  000200                                  MOV     #128.,R1                ;SET EXPECTATIONS:
                                                                                ;CALC # OF DATA PACKETS TO EXPECT:
022360  012703  000034                                  MOV     #XSFLG,R3       ;GET OFFSET
022364  060503                                          ADD     R5,R3           ;ABS. ADDR. OF XSFLG
022366  005002                                          CLR     R2              ;PRESET AS NONE
022370  005202                                  78$:    INC     R2              ;# PACKETS EXPECTED
022372  012723  000001                                  MOV     #RSDATA,(R3)+   ;LOAD XSFLG
022376  012723  000204                                  MOV     #132.,(R3)+     ;AND EXPECTED COUNT
022402  162701  000200                                  SUB     #128.,R1        ;NEG RESULT LAST TIME
022406  101401                                          BLOS    80$             ;LAST TIME
022410  000767                                          BR      78$             ;MORE TO DO
022412  005202                                  80$:    INC     R2              ;ADD ONE FOR END PACK
022414  010265  000032                                  MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
022420  012723  000002                                  MOV     #RSEND,(R3)+    ;EXPECT AN END ALSO...
022424  012713  000016                                  MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES
022430  004737  006560                                  CALL    RSVP            ;SEND
                                                                                ;AND RETURN TO SCHEDULER
```

```
022434  032715  002010                  81$:    BIT     #BIT10!BIT3,@R5 ;RETRY?
022440  001500                                  BEQ     79$             ;NO.
022442                                          .URTRY  REC(R5),#128.,DR(R5)    ;YES


022442  012700  027722                  86$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
022446  112710  000002                          MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
022452  112760  000012  000001                  MOVB    #RSMSIZ,1(R0)   ;THIS BIG
022460  112760  000002  000002                  MOVB    #RSSRD,2(R0)    ;OP CODE-READ
022466  016560  000064  000012                  MOV     REC(R5),10.(R0) ;THIS RECORD
022474  116560  000060  000004                  MOVB    DR(R5),4.(R0)   ;THIS DRIVE
022502  105060  000003                          CLRB    3(R0)           ;PRESET NORM THRESHOLD
022506  105715                                  TSTB    @R5             ;REDUCED?
022510  100002                                  BPL     87$             ;NO
022512  105260  000003                          INCB    3(R0)           ;YES-CHANGE THRESHOLD
022516  012760  000200  000010          87$:    MOV     #128.,8.(R0)    ;# BYTES DESIRED
022524  112760  000020  000005                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE
022532  005060  000006                          CLR     6.(R0)          ;NO SEQUENCE #
022536  012701  000012                          MOV     #RSMSIZ,R1      ;SIZE OF PACKET
022542  005721                                  TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
022544  012765  000016  000070                  MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND

022552  004737  013654                          CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
022556  010110                                  MOV     R1,(R0)         ;INSERT IN PACKET

022560  012701  000200                          MOV     #128.,R1                ;SET EXPECTATIONS:
                                                                        ;CALC # OF DATA PACKETS TO EXPECT
022564  012703  000034                          MOV     #XSFLG,R3       ;OFFSET OF FLAG
022570  060503                                  ADD     R5,R3           ;ABS. ADDR. OF XSFLG
022572  005002                                  CLR     R2              ;PRESET
022574  005202                          83$:    INC     R2              ;# PACKETS EXPECTED
022576  012723  000001                          MOV     #RSDATA,(R3)+   ;LOAD XSFLG
022602  012723  000204                          MOV     #132.,(R3)+     ;AND EXPECT COUNT
022606  162701  000200                          SUB     #128.,R1        ;NEG RESULT LAST TIME
022612  101401                                  BLOS    85$             ;LAST TIME!
022614  000767                                  BR      83$             ;MORE TO DO
022616  005202                          85$:    INC     R2              ;ADD ONE FOR END PACK
022620  010265  000032                          MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
022624  012723  000002                          MOV     #RSEND,(R3)+    ;EXPECT AN END
022630  012713  000016                          MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES

022634  004737  006560                          CALL    RSVP            ;SEND
                                                                        ;AND RETURN TO SCHEDULER

3668 022644  062704  000002                  ADD     #2,R4           ;POINT TO NEXT DATA
3669 022650  005764  022742                  TST     TST3PT(R4)      ;END?
3670 022654  001402                          BEQ     2$              ;YES
3671 022656  000137  021432                  JMP     1$              ;NO-WRITE, READ NEW DATA
3672 022662  005004                  2$:     CLR     R4              ;POINT TO FIRST DATA
3673 022664  062765  001000  000064          ADD     #1000,REC(R5)   ;BUT NOW USE ADJACENT RECORD
3674 022672  032765  004000  000064          BIT     #4000,REC(R5)   ;ALL ADJACENT RECORDS DONE?
3675 022700  001002                          BNE     3$              ;YES
3676 022702  000137  021432                  JMP     1$              ;NO-WRITE, READ AT NEW RECORD
3677 022706  162765  004000  000064  3$:     SUB     #4000,REC(R5)   ;RESTORE TO NEXT RECORD
3678 022714  066565  000066  000064          ADD     TMP(R5),REC(R5) ;HALF INTO REST OF TAPE
3679 022722  006265  000066                  ASR     TMP(R5)         ;HALF OF HALF FOR NEXT TIME
3680 022726  103402                          BCS     4$              ;DONE?
```

```
 3681 022730  000137  021432                    JMP     1$              ;NO
 3682 022734  005237  003322          4$:       INC     DONE            ;YES-SET FLAG
 3683 022740  000207                            RETURN
 3684 022742  000000                  TST3PT:   .WORD   000000
 3685 022744  125252                            .WORD   125252
 3686 022746  177777                            .WORD   177777
 3687 022750  052525                            .WORD   052525
 3688 022752  000000                            .WORD   000000
 3689
 3690
 3691 022754                                    ENDTST
      022754                                                    L10017:
      022754  104401                                                    TRAP    C$ETST
```

```
3694
3695                                    .SBTTL  TEST 5 / WRITE SELECTED NUMBER OF BLOCKS
3696
3697 022756                                     BGNTST
     022756                                                                                 T5::
3698 022756                                     ISTI)   #TST5
     022756  012737  023022  003326                                    MOV     #TST5,TSTTOP    ;SAVE ADDR OF TEST
     022764  004737  006004                                            CALL    SETUP           ;INIT UNITS TSTPC
     022770  004737  005632                                            CALL    SETDR           ;GET 1ST DRVS.
     022774  004737  006052                                            CALL    RUN             ;DO TEST
     023000  004737  005530                                            CALL    SWAPDR          ;GET NEXT DRVS.
     023004  103004                                                    BCC     64$             ;BR NO 2ND DRVS
     023006  004737  006004                                            CALL    SETUP           ;REINIT UNITS TSTPC
     023012  004737  006052                                            CALL    RUN             ;REPEAT TEST
     023016                                                  64$:                              ;DONE
3699 023016                                     EXIT TST
     023016  104432                                                                                     TRAP    C$EXIT
     023020  000724                                                                                     .WORD   L10020-.
3700
3701
3702 023022  005065  000064            TST5:   CLR     REC(R5)         ;START AT REC 0
3703 023026  013765  003310  000066            MOV     TAPLEN,TMP(R5)  ;GET THE # OF BLOCKS PER TRACK
3704 023034  005065  000062                    CLR     TRK(R5)         ;TRK(R5)=1ST OR 2ND PASS COUNTER
3705 023040  016565  000064  000072    1$:     MOV     REC(R5),PATTEN(R5)  ;USE RECORD NO. FOR DATA
3706 023046  005737  002220                    TST     DRVCHK                  ;ADD DR #?
3707 023052  001403                            BEQ     10$                     ;NO
3708 023054  066565  000060  000072            ADD     DR(R5),PATTEN(R5)       ;YES, ADD DRIVE ID
3709 023062                            10$:    TUWRIT  PATTEN(R5),REC(R5),#512.,DR(R5),#0
     023062  012700  027722                                    72$:    MOV     #TRBUF,R0       ;MAKE COMMAND PACKET:
     023066  112710  000002                                            MOVB    #RSCMND,@R0     ;COMMAND FLAG
     023072  112760  000012  000001                                    MOVB    #RSMSIZ,1(R0)   ;THIS SIZE
     023100  112760  000003  000002                                    MOVB    #RSSWR,2(R0)    ;INSERT OP CODE-WRITE
     023106  112760  000000  000003                                    MOVB    #0,3.(R0)       ;VERIFY (1 OR 0)
     023114  116560  000060  000004                                    MOVB    DR(R5),4.(R0)   ;DRIVE #
     023122  112760  000020  000005                                    MOVB    #020,5.(R0)     ;MAINTENANCE MODE SWITCH
     023130  005060  000006                                            CLR     6.(R0)          ;NO SEQUENCE #
     023134  012760  001000  000010                                    MOV     #512.,8.(R0)    ;TOTAL COUNT TO WRITE
     023142  016560  000064  000012                                    MOV     REC(R5),10.(R0) ;AT  RECORD N
     023150  012701  000012                                            MOV     #RSMSIZ,R1      ;THE PACKET SIZE PLUS+2
     023154  005721                                                    TST     (R1)+           ;(FLAG AND COUNT) INTO R1
     023156  012765  000016  000070                                    MOV     #RSSNSZ,SNDCNT(R5)  ;LOAD THE  SIZE TO SEND
     023164  004737  013654                                            CALL    CHKSUM          ;R0 --> R1=COUNT
     023170  010110                                                    MOV     R1,(R0)         ;PUT CHKSUM IN PACKET
                                                                                               ;SET UP EXPECTATIONS:
     023172  012765  000020  000034                                    MOV     #RSCONT,XSFLG(R5)   ;THE FLAG
     023200  012765  000001  000036                                    MOV     #1,XSCNT(R5)    ;THE COUNT
     023206  012765  000001  000032                                    MOV     #1,XSPKNM(R5)   ;THE # PACKETS EXPECTED
     023214  012702  001000                                            MOV     #512.,R2                ;GET # OF DATA BYTES
     023220  004737  006560                                            CALL    RSVP            ;SEND (AND RETURN TO SCHEDULER)
     023224  032715  000010                                            BIT     #BIT3,@R5       ;FLAG BYTE ERROR?
     023230  001314                                                    BNE     72$             ;YES
     023232  042715  010000                                            BIC     #BIT12,@R5      ;FLAG FOR LAST PACKET
     023236  012700  027722                                    64$:    MOV     #TRBUF,R0       ;POINT TO TOP OF BUFFER AGAIN
     023242  020227  000200                                            CMP     R2,#128.        ;START DATA PACKET(S)
     023246  101004                                                    BHI     65$             ;#512. > 128.!
     023250  010201                                                    MOV     R2,R1           ;#512.<128.
     023252  052715  010000                                            BIS     #BIT12,@R5      ;SO LAST PACKET NOW
```

```
023256  000402                                          BR      66$             ;USE REMAINING COUNT
023260  012701  000200                          65$:    MOV     #128.,R1        ;USE 128. BYTES
023264  110160  000001                          66$:    MOVB    R1,1(R0)        ;COPY COUNT TO BUFFER
023270  010103                                          MOV     R1,R3           ;R3=COUNTER TO LOAD BUFFER
023272  112710  000001                                  MOVB    #RSDATA,@R0     ;FLAG FIRST
023276  005720                                          TST     (R0)+           ;SKIP COUNT
023300  116520  000072                          67$:    MOVB    PATTEN(R5),(R0)+        ;INSERT DATA
023304  005303                                          DEC     R3              ;MORE?
023306  101374                                          BHI     67$             ;YES
023310  012700  027722                                  MOV     #TRBUF,R0       ;-->TOP AGAIN
023314  116001  000001                                  MOVB    1(R0),R1        ;GET COUNT
023320  042701  177400                                  BIC     #177400,R1      ;ZERO SIGN EXTEND
023324  010165  000070                                  MOV     R1,SNDCNT(R5)   ;HOW MANY TO SEND PLUS
023330  062765  000004  000070                         ADD     #4,SNDCNT(R5)   ;FLAG,COUNT,CHKSUM
023336  062701  000002                                  ADD     #2,R1           ;COMPENSATE FOR FLAG + COUNT
023342  004737  013654                                  CALL    CHKSUM          ;FOR CHECKSUM CALC.
023346  110120                                          MOVB    R1,(R0)+        ;CHKSUM INTO PACKET
023350  000301                                          SWAB    R1              ;EVEN ON AN ODD
023352  110120                                          MOVB    R1,(R0)+        ;BYTE BOUNDARY
023354  032715  010000                                  BIT     #BIT12,@R5      ;LAST DATA PACKET?
023360  001412                                          BEQ     68$             ;NO
023362  012765  000002  000034                         MOV     #RSEND,XSFLG(R5)  ;YES-EXPECT 'END'
023370  012765  000016  000036                         MOV     #RSNDSZ,XSCNT(R5)  ;OF THIS SIZE
023376  012765  000001  000032                         MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
023404  000411                                          BR      69$             ;SEND
023406  012765  000020  000034          68$:    MOV     #RSCONT,XSFLG(R5)  ;(NOT LAST), EXPECT 'CONTINUE'
023414  012765  000001  000036                         MOV     #1,XSCNT(R5)    ;AND 1 BYTE
023422  012765  000001  000032                         MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
023430  004737  006560                          69$:    CALL    RSVP            ;SEND PACKET
                                                                                ;AND RETURN TO SCHEDULER
023434  032715  000010                                  BIT     #3IT3,@R5       ;FLAG BYTE RETRY?
023440  001210                                          BNE     72$             ;YES
023442  032715  002000                                  BIT     #BIT10,@R5      ;RETRY DATA ERROR?
023446  001004                                          BNE     70$             ;YES
023450  162702  000200                                  SUB     #128.,R2        ;NO, MORE DATA TO SEND?
023454  101270                                          BHI     64$             ;YES
023456  000502                                          BR      71$             ;NO
023460                                          70$:    TURTRY  REC(R5),#512.,DR(R5)    ;RETRY HERE


023460  012700  027722                          76$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
023464  112710  000002                                  MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
023470  112760  000012  000001                         MOVB    #RSMSIZ,1(R0)   ;THIS BIG
023476  112760  000002  000002                         MOVB    #RSSRD,2(R0)    ;OP CODE-READ
023504  016560  000064  000012                         MOV     REC(R5),10.(R0) ;THIS RECORD
023512  116560  000060  000004                         MOVB    DR(R5),4.(R0)   ;THIS DRIVE
023520  105060  000003                                  CLRB    3(R0)           ;PRESET NORM THRESHOLD
023524  105715                                          TSTB    @R5             ;REDUCED?
023526  100002                                          BPL     77$             ;NO
023530  105260  000003                                  INCB    3(R0)           ;YES-CHANGE THRESHOLD
023534  012760  001000  000010          77$:    MOV     #512.,8.(R0)    ;# BYTES DESIRED
023542  112760  000020  000005                         MOVB    #020,5.(R0)     ;MAINTENANCE MODE
023550  005060  000006                                  CLR     6.(R0)          ;NO SEQUENCE #
023554  012701  000012                                  MOV     #RSMSIZ,R1      ;SIZE OF PACKET
023560  005721                                          TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
023562  012765  000016  000070                         MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND
```

```
        023570  004737  013654                          CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
        023574  010110                                   MOV     R1,(R0)         ;INSERT IN PACKET

        023576  012701  001000                          MOV     #512.,R1                ;SET EXPECTATIONS:
                                                                                 ;CALC # OF DATA PACKETS TO EXPECT
        023602  012703  000034                          MOV     #XSFLG,R3       ;OFFSET OF FLAG
        023606  060503                                   ADD     R5,R3           ;ABS. ADDR. OF XSFLG
        023610  005002                                   CLR     R2              ;PRESET
        023612  005202                          73$:     INC     R2              ;# PACKETS EXPECTED
        023614  012723  000001                          MOV     #RSDATA,(R3)+   ;LOAD XSFLG
        023620  012723  000204                          MOV     #132.,(R3)+     ;AND EXPECT COUNT
        023624  162701  000200                          SUB     #128.,R1        ;NEG RESULT LAST TIME
        023630  101401                                   BLOS    75$             ;LAST TIME!
        023632  000767                                   BR      73$             ;MORE TO DO
        023634  005202                          75$:     INC     R2              ;ADD ONE FOR END PACK
        023636  010265  000032                          MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
        023642  012723  000002                          MOV     #RSEND,(R3)+    ;EXPECT AN END
        023646  012713  000016                          MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES

        023652  004737  006560                          CALL    RSVP            ;SEND
                                                                                 ;AND RETURN TO SCHEDULER

3710    023666  005365  000066                  DEC     TMP(R5)         ;DO ALL RECORDS FOR THIS TRACK?
3711    023672  001404                          BEQ     2$              ;YES-GET OTHER TRACK
3712    023674  005265  000064                  INC     REC(R5)         ;NO-ONTO NEXT RECORD
3713    023700  000137  023040                  JMP     1$              ;EXECUTE THE WRITE
3714    023704  005765  000062          2$:     TST     TRK(R5)         ;DONE 2 TRACKS?
3715    023710  001012                          BNE     TST5EX          ;YES-EXIT
3716    023712  005265  000062                  INC     TRK(R5)         ;NO-SET FLAG FOR NEXT PASS
3717    023716  013765  003334  000064          MOV     SECREC,REC(R5)  ;GET NEW STARTING BLOCK #
3718    023724  013765  003310  000066          MOV     TAPLEN,TMP(R5)  ;RESET # OF BLOCKS
3719    023732  000137  023040                  JMP     1$              ;AND EXECUTE
3720    023736  005237  003322          TST5EX: INC     DONE            ;DONE
3721    023742  000207                          RETURN                  ;RETURN
3722
3723    023744                                  ENDTST
        023744                                                                  L10020:
        023744  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  TEST 6 / READ SELECTED NUMBER OF BLOCKS

                                        BGNTST                                  T6::
                                        TSTID   #TST6
023746 012737 024012 003326                             MOV     #TST6,TSTTOP    ;SAVE ADDR OF TEST
023754 004737 006004                    CALL    SETUP           ;INIT UNITS TSTPC
023760 004737 005632                    CALL    SETDR           ;GET 1ST DRVS.
023764 004737 006052                    CALL    RUN             ;DO TEST
023770 004737 005530                    CALL    SWAPDR          ;GET NEXT DRVS.
023774 103004                           BCC     64$             ;BR NO 2ND DRVS
023776 004737 006004                    CALL    SETUP           ;REINIT UNITS TSTPC
024002 004737 006052                    CALL    RUN             ;REPEAT TEST
024006                          64$:                            ;DONE
                                        EXIT TST
024006 104432                                                           TRAP    C$EXIT
024010 000520                                                           .WORD   L10021-.

024012 005065 000064            TST6:   CLR     REC(R5)         ;START AT REC 0
024016 013765 003310 000066             MOV     TAPLEN,TMP(R5)  ;GET THE # OF BLOCKS PER TRACK
024024 005065 000062                    CLR     TRK(R5)         ;TRK(R5)=1ST OR 2ND PASS
024030 016565 000064 000072     1$:     MOV     REC(R5),PATTEN(R5)  ;USE RECORD NO. AS DATA
024036 005737 002220                    TST     DRVCHK          ;ADD DR #?
024042 001403                           BEQ     10$             ;NO
024044 066565 000060 000072             ADD     DR(R5),PATTEN(R5)   ;ADD IN DRIVE ID
024052                          10$:    TUREAD  REC(R5),#512.,DR(R5),#0


024052 012700 027722                    MOV     #TRBUF,R0       ;FORM CMND PACK:
024056 112710 000002                    MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
024062 112760 000012 000001             MOVB    #RSMSIZ,1(R0)   ;THIS BIG
024070 112760 000002 000002             MOVB    #RSSRD,2(R0)    ;OP CODE IS READ
024076 016560 000064 000012             MOV     REC(R5),10.(R0) ;THIS RECORD
024104 116560 000060 000004             MOVB    DR(R5),4.(R0)   ;THIS DRIVE
024112 112760 000000 000003             MOVB    #0,3.(R0)       ;VERIFY
024120 012760 001000 000010             MOV     #512.,8.(R0)    ;TOTAL BYTES TO READ
024126 112760 000020 000005             MOVB    #020,5.(R0)     ;MAINTENANCE MODE
024134 005060 000006                    CLR     6.(R0)          ;NO SEQUENCE #
024140 012701 000012                    MOV     #RSMSIZ,R1      ;GET SIZE OF PACKET
024144 005721                           TST     (R1)+           ;+2 FOR CHECKSUM
024146 012765 000016 000070             MOV     #RSSNSZ,SNDCNT(R5)  ;SIZE TO SEND
024154 004737 013654                    CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
024160 010110                           MOV     R1,(R0)         ;INSERT CHECKSUM

024162 012701 001000                    MOV     #512.,R1                ;SET EXPECTATIONS:
                                                                ;CALC # OF DATA PACKETS TO EXPECT:
024166 012703 000034                    MOV     #XSFLG,R3       ;GET OFFSET
024172 060503                           ADD     R5,R3           ;ABS. ADDR. OF XSFLG
024174 005002                           CLR     R2              ;PRESET AS NONE
024176 005202                   64$:    INC     R2              ;# PACKETS EXPECTED
024200 012723 000001                    MOV     #RSDATA,(R3)+   ;LOAD XSFLG
024204 012723 000204                    MOV     #132.,(R3)+     ;AND EXPECTED COUNT
024210 162701 000200                    SUB     #128.,R1        ;NEG RESULT LAST TIME
024214 101401                           BLOS    66$             ;LAST TIME
024216 000767                           BR      64$             ;MORE TO DO
024220 005202                   66$:    INC     R2              ;ADD ONE FOR END PACK
```

```
024222  010265  000032                          MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
024226  012723  000002                          MOV     #RSEND,(R3)+    ;EXPECT AN END ALSO...
024232  012713  000016                          MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES
024236  004737  006560                          CALL    RSVP            ;SEND
                                                                        ;AND RETURN TO SCHEDULER
024242  032715  002010                  67$:    BIT     #BIT10!BIT3,@R5 ;RETRY?
024246  001500                                  BEQ     65$             ;NO.
024250                                          TURTRY  REC(R5),#512.,DR(R5)    ;YES


024250  012700  027722                  72$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
024254  112710  000002                          MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
024260  112760  000012  000001                  MOVB    #RSMSIZ,1(R0)   ;THIS BIG
024266  112760  000002  000002                  MOVB    #RSSRD,2(R0)    ;OP CODE-READ
024274  016560  000064  000012                  MOV     REC(R5),10.(R0) ;THIS RECORD
024302  116560  000060  000004                  MOVB    DR(R5),4.(R0)   ;THIS DRIVE
024310  105060  000003                          CLRB    3(R0)           ;PRESET NORM THRESHOLD
024314  105715                                  TSTB    @R5             ;REDUCED?
024316  100002                                  BPL     73$             ;NO
024320  105260  000003                          INCB    3(R0)           ;YES-CHANGE THRESHOLD
024324  012760  001000  000010          73$:    MOV     #512.,8.(R0)    ;# BYTES DESIRED
024332  112760  000020  000005                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE
024340  005060  000006                          CLR     6.(R0)          ;NO SEQUENCE #
024344  012701  000012                          MOV     #RSMSIZ,R1      ;SIZE OF PACKET
024350  005721                                  TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
024352  012765  000016  000070                  MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND

024360  004737  013654                          CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
024364  010110                                  MOV     R1,(R0)         ;INSERT IN PACKET

024366  012701  001000                          MOV     #512.,R1                ;SET EXPECTATIONS:
                                                                        ;CALC # OF DATA PACKETS TO EXPECT
024372  012703  000034                          MOV     #XSFLG,R3       ;OFFSET OF FLAG
024376  060503                                  ADD     R5,R3           ;ABS. ADDR. OF XSFLG
024400  005002                                  CLR     R2              ;PRESET
024402  005202                          69$:    INC     R2              ;# PACKETS EXPECTED
024404  012723  000001                          MOV     #RSDATA,(R3)+   ;LOAD XSFLG
024410  012723  000204                          MOV     #132.,(R3)+     ;AND EXPECT COUNT
024414  162701  000200                          SUB     #128.,R1        ;NEG RESULT LAST TIME
024420  101401                                  BLOS    71$             ;LAST TIME!
024422  000767                                  BR      69$             ;MORE TO DO
024424  005202                          71$:    INC     R2              ;ADD ONE FOR END PACK
024426  010265  000032                          MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
024432  012723  000002                          MOV     #RSEND,(R3)+    ;EXPECT AN END
024436  012713  000016                          MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES

024442  004737  006560                          CALL    RSVP            ;SEND
                                                                        ;AND RETURN TO SCHEDULER

3741 024452  005365  000066                  DEC     TMP(R5)         ;DO ALL RECORDS THIS TRACK?
3742 024456  001404                          BEQ     2$              ;YES-GET OTHER TRACK
3743 024460  005265  000064                  INC     REC(R5)         ;NO-NEXT RECORD
3744 024464  000137  024030                  JMP     1$              ;EXECUTE THE READ
3745 024470  005765  000062          2$:     TST     TRK(R5)         ;DONE 2 TRACKS?
3746 024474  001012                          BNE     TST6EX          ;YES-EXIT
3747 024476  005265  000062                  INC     TRK(R5)         ;NO-SET FLAG FOR NEXT PASS
3748 024502  013765  003334  000064          MOV     SECREC,REC(R5)  ;GET NEW STARTING BLOCK #
```

```
 3749 024510  013765  003310  000066          MOV     TAPLEN,TMP(R5)  ;RESET # OF BLOCKS
 3750 024516  000137  024030                  JMP     1$              ;AND EXECUTE
 3751 024522  005237  003322          TST6EX: INC     DONE            ;DONE
 3752 024526  000207                          RETURN                  ;RETURN
 3753
 3754 024530                                  ENDTST
      024530                                                          L10021:
      024530  104401                                                          TRAP    C$ETST
```

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

                                        BGNTST
                                                                                T7::
                                        TSTID   #TST7
024532  012737  024576  003326                        MOV     #TST7,TSTTOP    ;SAVE ADDR OF TEST
024540  004737  006004                                CALL    SETUP           ;INIT UNITS TSTPC
024544  004737  005632                                CALL    SETDR           ;GET 1ST DRVS.
024550  004737  006052                                CALL    RUN             ;DO TEST
024554  004737  005530                                CALL    SWAPDR          ;GET NEXT DRVS.
024560  103004                                        BCC     64$             ;BR NO 2ND DRVS
024562  004737  006004                                CALL    SETUP           ;REINIT UNITS TSTPC
024566  004737  006052                                CALL    RUN             ;REPEAT TEST
024572                                          64$:                          ;DONE
                                        EXIT TST
024572  104432                                                          TRAP    C$EXIT
024574  000724                                                          .WORD   L10022-.

024576  005065  000064          TST7:   CLR     REC(R5)                 ;START AT REC 0
024602  013765  003310  000066          MOV     TAPLEN,TMP(R5)          ;GET THE # OF BLOCKS PER TRACK
024610  005065  000062                  CLR     TRK(R5)                 ;TRK(R5)=1ST OR 2ND PASS
024614  016565  000064  000072  1$:     MOV     REC(R5),PATTEN(R5)      ;USE RECORD NO. FOR DATA
024622  005737  002220                  TST     DRVCHK                  ;ADD DR #?
024626  001403                          BEQ     10$                     ;NO
024630  066565  000060  000072          ADD     DR(R5),PATTEN(R5)               ;ADD DRIVE ID
024636                          10$:    TUWRIT  PATTEN(R5),REC(R5),#512.,DR(R5),#1
                                                72$:    MOV     #TRBUF,R0       ;MAKE COMMAND PACKET:
024636  012700  027722                          MOVB    #RSCMND,@R0     ;COMMAND FLAG
024642  112710  000002                          MOVB    #RSMSIZ,1(R0)   ;THIS SIZE
024646  112760  000012  000001                  MOVB    #RSSWR,2(R0)    ;INSERT OP CODE-WRITE
024654  112760  000003  000002                  MOVB    #1,3.(R0)       ;VERIFY (1 OR 0)
024662  112760  000001  000003                  MOVB    DR(R5),4.(R0)   ;DRIVE #
024670  116560  000060  000004                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE SWITCH
024676  112760  000020  000005                  CLR     6.(R0)          ;NO SEQUENCE #
024704  005060  000006                          MOV     #512.,8.(R0)    ;TOTAL COUNT TO WRITE
024710  012760  001000  000010                  MOV     REC(R5),10.(R0) ;AT  RECORD N
024716  016560  000064  000012                  MOV     #RSMSIZ,R1      ;THE PACKET SIZE PLUS+2
024724  012701  000012                          TST     (R1)+           ;(FLAG AND COUNT) INTO R1
024730  005721                                  MOV     #RSSNSZ,SNDCNT(R5)      ;LOAD THE  SIZE TO SEND
024732  012765  000016  000070                  CALL    CHKSUM          ;R0 --> R1=COUNT
024740  004737  013654                          MOV     R1,(R0)         ;PUT CHKSUM IN PACKET
024744  010110                                                          ;SET UP EXPECTATIONS:

024746  012765  000020  000034                  MOV     #RSCONT,XSFLG(R5)       ;THE FLAG
024754  012765  000001  000036                  MOV     #1,XSCNT(R5)    ;THE COUNT
024762  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;THE # PACKETS EXPECTED
024770  012702  001000                          MOV     #512.,R2                ;GET # OF DATA BYTES
024774  004737  006560                          CALL    RSVP            ;SEND (AND RETURN TO SCHEDULER)
025000  032715  000010                          BIT     #BIT3,@R5       ;FLAG BYTE ERROR?
025004  001314                                  BNE     72$             ;YES
025006  042715  010000                          BIC     #BIT12,@R5      ;FLAG FOR LAST PACKET
025012  012700  027722                  64$:    MOV     #TRBUF,R0       ;POINT TO TOP OF BUFFER AGAIN
025016  020227  000200                          CMP     R2,#128.        ;START DATA PACKET(S)
025022  101004                                  BHI     65$             ;#512. > 128.!
025024  010201                                  MOV     R2,R1           ;#512.<128.
025026  052715  010000                          BIS     #BIT12,@R5      ;SO LAST PACKET NOW
025032  000402                                  BR      66$             ;USE REMAINING COUNT
```

```
025034  012701  000200                  65$:    MOV     #128.,R1        ;USE 128. BYTES
025040  110160  000001                  66$:    MOVB    R1,1(R0)        ;COPY COUNT TO BUFFER
025044  010103                                  MOV     R1,R3           ;R3=COUNTER TO LOAD BUFFER
025046  112710  000001                          MOVB    #RSDATA,@R0     ;FLAG FIRST
025052  005720                                  TST     (R0)+           ;SKIP COUNT
025054  116520  000072                  67$:    MOVB    PATTEN(R5),(R0)+        ;INSERT DATA
025060  005303                                  DEC     R3              ;MORE?
025062  101374                                  BHI     67$             ;YES
025064  012700  027722                          MOV     #TRBUF,R0       ;-->TOP AGAIN
025070  116001  000001                          MOVB    1(R0),R1        ;GET COUNT
025074  042701  177400                          BIC     #177400,R1      ;ZERO SIGN EXTEND
025100  010165  000070                          MOV     R1,SNDCNT(R5)   ;HOW MANY TO SEND PLUS
025104  062765  000004  000070                  ADD     #4,SNDCNT(R5)   ;FLAG,COUNT,CHKSUM
025112  062701  000002                          ADD     #2,R1           ;COMPENSATE FOR FLAG + COUNT
025116  004737  013654                          CALL    CHKSUM          ;FOR CHECKSUM CALC.
025122  110120                                  MOVB    R1,(R0)+        ;CHKSUM INTO PACKET
025124  000301                                  SWAB    R1              ;EVEN ON AN ODD
025126  110120                                  MOVB    R1,(R0)+        ;BYTE BOUNDARY
025130  032715  010000                          BIT     #BIT12,@R5      ;LAST DATA PACKET?
025134  001412                                  BEQ     68$             ;NO
025136  012765  000002  000034                  MOV     #RSEND,XSFLG(R5)  ;YES-EXPECT 'END'
025144  012765  000016  000036                  MOV     #RSNDSZ,XSCNT(R5)  ;OF THIS SIZE
025152  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
025160  000411                                  BR      69$             ;SEND
025162  012765  000020  000034          68$:    MOV     #RSCONT,XSFLG(R5)  ;(NOT LAST), EXPECT 'CONTINUE'
025170  012765  000001  000036                  MOV     #1,XSCNT(R5)    ;AND 1 BYTE
025176  012765  000001  000032                  MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
025204  004737  006560                  69$:    CALL    RSVP            ;SEND PACKET
                                                                        ;AND RETURN TO SCHEDULER
025210  032715  000010                          BIT     #BIT3,@R5       ;FLAG BYTE RETRY?
025214  001210                                  BNE     72$             ;YES
025216  032715  002000                          BIT     #BIT10,@R5      ;RETRY DATA ERROR?
025222  001004                                  BNE     70$             ;YES
025224  162702  000200                          SUB     #128.,R2        ;NO, MORE DATA TO SEND?
025230  101270                                  BHI     64$             ;YES
025232  000502                                  BR      71$             ;NO
025234                          70$:    TURTRY  REC(R5),#512.,DR(R5)    ;RETRY HERE


025234  012700  027722                  76$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
025240  112710  000002                          MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
025244  112760  000012  000001                  MOVB    #RSMSIZ,1(R0)   ;THIS BIG
025252  112760  000002  000002                  MOVB    #RSSRD,2(R0)    ;OP CODE-READ
025260  016560  000064  000012                  MOV     REC(R5),10.(R0) ;THIS RECORD
025266  116560  000060  000004                  MOVB    DR(R5),4.(R0)   ;THIS DRIVE
025274  105060  000003                          CLRB    3(R0)           ;PRESET NORM THRESHOLD
025300  105715                                  TSTB    @R5             ;REDUCED?
025302  100002                                  BPL     77$             ;NO
025304  105260  000003                          INCB    3(R0)           ;YES-CHANGE THRESHOLD
025310  012760  001000  000010          77$:    MOV     #512.,8.(R0)    ;# BYTES DESIRED
025316  112760  000020  000005                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE
025324  005060  000006                          CLR     6.(R0)          ;NO SEQUENCE #
025330  012701  000012                          MOV     #RSMSIZ,R1      ;SIZE OF PACKET
025334  005721                                  TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
025336  012765  000016  000070                  MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND

025344  004737  013654                          CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
```

```
025350  010110                                          MOV     R1,(R0)         ;INSERT IN PACKET

025352  012701  001000                                  MOV     #512.,R1                ;SET EXPECTATIONS:
                                                                                ;CALC # OF DATA PACKETS TO EXPECT
025356  012703  000034                                  MOV     #XSFLG,R3       ;OFFSET OF FLAG
025362  060503                                          ADD     R5,R3           ;ABS. ADDR. OF XSFLG
025364  005002                                          CLR     R2              ;PRESET
025366  005202                                  73$:    INC     R2              ;# PACKETS EXPECTED
025370  012723  000001                                  MOV     #RSDATA,(R3)+   ;LOAD XSFLG
025374  012723  000204                                  MOV     #132.,(R3)+     ;AND EXPECT COUNT
025400  162701  000200                                  SUB     #128.,R1        ;NEG RESULT LAST TIME
025404  101401                                          BLOS    75$             ;LAST TIME!
025406  000767                                          BR      73$             ;MORE TO DO
025410  005202                                  75$:    INC     R2              ;ADD ONE FOR END PACK
025412  010265  000032                                  MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
025416  012723  000002                                  MOV     #RSEND,(R3)+    ;EXPECT AN END
025422  012713  000016                                  MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES

025426  004737  006560                                  CALL    RSVP            ;SEND
                                                                                ;AND RETURN TO SCHEDULER

 3772 025442  005365  000066                    DEC     TMP(R5)         ;DO ALL RECORDS FOR THIS TRACK?
 3773 025446  001404                            BEQ     2$              ;YES-GET OTHER TRACK
 3774 025450  005265  000064                    INC     REC(R5)         ;NO-NEXT RECORD
 3775 025454  000137  024614                    JMP     1$              ;EXECUTE THE WRITE
 3776 025460  005765  000062            2$:     TST     TRK(R5)         ;DONE 2 TRACKS?
 3777 025464  001012                            BNE     TST7EX          ;YES-EXIT
 3778 025466  005265  000062                    INC     TRK(R5)         ;NO-SET FLAG FOR NEXT PASS
 3779 025472  013765  003334  000064            MOV     SECREC,REC(R5)  ;GET NEW STARTING BLOCK #
 3780 025500  013765  003310  000066            MOV     TAPLEN,TMP(R5)  ;RESET # OF BLOCKS
 3781 025506  000137  024614                    JMP     1$              ;AND EXECUTE
 3782 025512  005237  003322            TST7EX: INC     DONE            ;DONE
 3783 025516  000207                            RETURN                  ;RETURN
 3784
 3785 025520                                    ENDTST
      025520                                                                    L10022:
      025520  104401                                                                   TRAP    CSETST
```

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

                                        BGNTST
                                                                                T8::
                                        TSTID   #TST8
025522 012737 025566 003326                             MOV     #TST8,TSTTOP    ;SAVE ADDR OF TEST
025530 004737 006004                                    CALL    SETUP           ;INIT UNITS TSTPC
025534 004737 005632                                    CALL    SETDR           ;GET 1ST DRVS.
025540 004737 006052                                    CALL    RUN             ;DO TEST
025544 004737 005530                                    CALL    SWAPDR          ;GET NEXT DRVS.
025550 103004                                           BCC     64$             ;BR NO 2ND DRVS
025552 004737 006004                                    CALL    SETUP           ;REINIT UNITS TSTPC
025556 004737 006052                                    CALL    RUN             ;REPEAT TEST
025562                                          64$:                            ;DONE
                                        EXIT TST
025562 104432                                                           TRAP    C$EXIT
025564 000520                                                           .WORD   L10023-.

025566 005065 000064            TST8:   CLR     REC(R5)                 ;START AT REC 0
025572 013765 003310 000066             MOV     TAPLEN,TMP(R5)          ;GET THE # OF BLOCKS PER TRACK
025600 005065 000062                    CLR     TRK(R5)                 ;TRK(R5)=1ST OR 2ND PASS
025604 016565 000064 000072     1$:     MOV     REC(R5),PATTEN(R5)      ;USE RECORD NO. FOR DATA
025612 005737 002220                    TST     DRVCHK                  ;ADD DR #?
025616 001403                           BEQ      10$                    ;NO
025620 066565 000060 000072             ADD     DR(R5),PATTEN(R5)               ;ADD DRIVE ID
025626                          10$:    TUREAD  REC(R5),#512.,DR(R5),#1

025626 012700 027722                    MOV     #TRBUF,R0               ;FORM CMND PACK:
025632 112710 000002                    MOVB    #RSCMND,@R0             ;MESSAGE PACK TYPE
025636 112760 000012 000001             MOVB    #RSMSIZ,1(R0)           ;THIS BIG
025644 112760 000002 000002             MOVB    #RSSRD,2(R0)            ;OP CODE IS READ
025652 016560 000064 000012             MOV     REC(R5),10.(R0)         ;THIS RECORD
025660 116560 000060 000004             MOVB    DR(R5),4.(R0)           ;THIS DRIVE
025666 112760 000001 000003             MOVB    #1,3.(R0)               ;VERIFY
025674 012760 001000 000010             MOV     #512.,8.(R0)            ;TOTAL BYTES TO READ
025702 112760 000020 000005             MOVB    #020,5.(R0)             ;MAINTENANCE MODE
025710 005060 000006                    CLR     6.(R0)                  ;NO SEQUENCE #
025714 012701 000012                    MOV     #RSMSIZ,R1              ;GET SIZE OF PACKET
025720 005721                           TST     (R1)+                   ;+2 FOR CHECKSUM
025722 012765 000016 000070             MOV     #RSSNSZ,SNDCNT(R5)      ;SIZE TO SEND
025730 004737 013654                    CALL    CHKSUM                  ;FORM CHECKSUM R1=COUNT
025734 010110                           MOV     R1,(R0)                 ;INSERT CHECKSUM

025736 012701 001000                    MOV     #512.,R1                        ;SET EXPECTATIONS:
                                                                        ;CALC # OF DATA PACKETS TO EXPECT:
025742 012703 000034                    MOV     #XSFLG,R3               ;GET OFFSET
025746 060503                           ADD     R5,R3                   ;ABS. ADDR. OF XSFLG
025750 005002                           CLR     R2                      ;PRESET AS NONE
025752 005202                   64$:    INC     R2                      ;# PACKETS EXPECTED
025754 012723 000001                    MOV     #RSDATA,(R3)+           ;LOAD XSFLG
025760 012723 000204                    MOV     #132.,(R3)+             ;AND EXPECTED COUNT
025764 162701 000200                    SUB     #128.,R1                ;NEG RESULT LAST TIME
025770 101401                           BLOS    66$                     ;LAST TIME
025772 000767                           BR      64$                     ;MORE TO DO
025774 005202                   66$:    INC     R2                      ;ADD ONE FOR END PACK
```

```
026776  010265  000032                          MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
027002  012723  000002                          MOV     #RSEND,(R3)+    ;EXPECT AN END ALSO...
027006  012713  000016                          MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES
027012  004737  006560                          CALL    RSVP            ;SEND
                                                                        ;AND RETURN TO SCHEDULER
027016  032715  002010                  67$:    BIT     #BIT10!BIT3,@R5 ;RETRY?
027022  001500                                  BEQ     65$             ;NO.
027024                                          TURTRY  REC(R5),#512.,DR(R5)    ;YES


027024  012700  027722                  72$:    MOV     #TRBUF,R0       ;FORM CMND PACK:
027030  112710  000002                          MOVB    #RSCMND,@R0     ;MESSAGE PACK TYPE
027034  112760  000012  000001                  MOVB    #RSMSIZ,1.(R0)  ;THIS BIG
027042  112760  000002  000002                  MOVB    #RSSRD,2(R0)    ;OP CODE-READ
027050  016560  000064  000012                  MOV     REC(R5),10.(R0) ;THIS RECORD
027056  116560  000060  000004                  MOVB    DR(R5),4.(R0)   ;THIS DRIVE
027064  105060  000003                          CLRB    3(R0)           ;PRESET NORM THRESHOLD
027070  105715                                  TSTB    @R5             ;REDUCED?
027072  100002                                  BPL     73$             ;NO
027074  105260  000003                          INCB    3(R0)           ;YES-CHANGE THRESHOLD
027100  012760  001000  000010          73$:    MOV     #512.,8.(R0)    ;# BYTES DESIRED
027106  112760  000020  000005                  MOVB    #020,5.(R0)     ;MAINTENANCE MODE
027114  005060  000006                          CLR     6.(R0)          ;NO SEQUENCE #
027120  012701  000012                          MOV     #RSMSIZ,R1      ;SIZE OF PACKET
027124  005721                                  TST     (R1)+           ;PLUS FLAG+COUNT INTO R1
027126  012765  000016  000070                  MOV     #RSSNSZ,SNDCNT(R5)  ;SET UP SIZE TO SEND

027134  004737  013654                          CALL    CHKSUM          ;FORM CHECKSUM R1=COUNT
027140  010110                                  MOV     R1,(R0)         ;INSERT IN PACKET

027142  012701  001000                          MOV     #512.,R1                ;SET EXPECTATIONS:
                                                                        ;CALC # OF DATA PACKETS TO EXPECT
027146  012703  000034                          MOV     #XSFLG,R3       ;OFFSET OF FLAG
027152  060503                                  ADD     R5,R3           ;ABS. ADDR. OF XSFLG
027154  005002                                  CLR     R2              ;PRESET
027156  005202                          69$:    INC     R2              ;# PACKETS EXPECTED
027160  012723  000001                          MOV     #RSDATA,(R3)+   ;LOAD XSFLG
027164  012723  000204                          MOV     #132.,(R3)+     ;AND EXPECT COUNT
027170  162701  000200                          SUB     #128.,R1        ;NEG RESULT LAST TIME
027174  101401                                  BLOS    71$             ;LAST TIME!
027176  000767                                  BR      69$             ;MORE TO DO
027200  005202                          71$:    INC     R2              ;ADD ONE FOR END PACK
027202  010265  000032                          MOV     R2,XSPKNM(R5)   ;SAVE # PACKETS TO EXPECT
027206  012723  000002                          MOV     #RSEND,(R3)+    ;EXPECT AN END
027212  012713  000016                          MOV     #RSNDSZ,(R3)    ;THIS BIG-14. BYTES

027216  004737  006560                          CALL    RSVP            ;SEND
                                                                        ;AND RETURN TO SCHEDULER

3803 027226  005365  000066                  DEC     TMP(R5)         ;DO ALL RECORDS THIS TRACK?
3804 027232  001404                          BEQ     2$              ;YES-GET OTHER TRACK
3805 027234  005265  000064                  INC     REC(R5)         ;NO-NEXT RECORD
3806 027240  000137  025604                  JMP     1$              ;EXECUTE THE READ
3807 027244  005765  000062          2$:     TST     TRK(R5)         ;DONE 2 TRACKS?
3808 027250  001012                          BNE     TST8EX          ;YES-EXIT
3809 027252  005265  000062                  INC     TRK(R5)         ;NO-SET FLAG FOR NEXT PASS
3810 027256  013765  003334  000064          MOV     SECREC,REC(R5)  ;GET NEW STARTING BLOCK #
```

```
3811 026264  013765  003310  000066          MOV     TAPLEN,TMP(R5)  ;RESET # OF BLOCKS
3812 026272  000137  025604                  JMP     1$              ;AND EXECUTE
3813 026276  005237  003322          TST8EX: INC     DONE            ;DONE
3814 026302  000207                          RETURN                  ;RETURN
3815
3816 026304                                  ENDTST
     026304                                                  L10023:
     026304  104401                                                  TRAP    C$ETST
```

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

026306                                  BGNTST
026306                                                                  T9::
026306  012737  026330  003326          MOV     #TST9,TSTTOP    ;SAVE ADDR OF TEST
026314  004737  006004                  CALL    SETUP           ;INIT UNITS TSTPC
026320  004737  006052                  CALL    RUN             ;DO TEST
                                                                ;DONE

026324                                  EXIT TST
026324  104432                                                          TRAP    CSEXIT
026326  000662                                                          .WORD   L10024-.

026330  012737  000001  003342  TST9:   MOV     #1,TEST9        ;INDICATES 1ST PART OF TST 8
026336  012700  027722          64$:    MOV     #TRBUF,R0       ;FORM COMMAND PACKET
026342  112710  000002                  MOVB    #RSCMND,@R0     ;COMMAND FLAG
026346  112760  000012  000001          MOVB    #RSMSIZ,1(R0)   ;SIZE OF MESSAGE
026354  112760  000012  000002          MOVB    #RSSGET,2(R0)   ;GET CHARACTERISTICS
026362  105060  000003                  CLRB    3(R0)           ;NO MODIFIER.
026366  005060  000004                  CLR     4(R0)           ;NO DRIVE OR SWITCHES
026372  005060  000006                  CLR     6(R0)           ;NO SEQUENCE NUMBER
026376  005060  000010                  CLR     8.(R0)          ;NO BYTES
026402  005060  000012                  CLR     10.(R0)         ;NO RECORD #
026406  012701  000012                  MOV     #RSMSIZ,R1      ;GET SIZE
026412  005721                          TST     (R1)+           ;+2 FOR CHECKSUM
026414  012765  000016  000070          MOV     #RSSNSZ,SNDCNT(R5)  ;SIZE TO SEND
026422  004737  013654                  CALL    CHKSUM          ;FORM CHECKSUM
026426  010110                          MOV     R1,(R0)         ;INSERT INTO PACKET
026430  012765  000001  000034          MOV     #RSDATA,XSFLG(R5)   ;EXPECT DATA PACKET
026436  012765  000034  000036          MOV     #RSGCDP,XSCNT(R5)   ;THIS BIG
026444  012765  000001  000032          MOV     #1,XSPKNM(R5)   ;AND 1 PACKET
                                                                ;SEND
026452  004737  006560                  CALL    RSVP            ;RETURN TO SCHEDULER

026456  004737  014010                  CALL    DOBRK           ;CLR POTENTIAL INTERFACE ERROR

026462  032715  000010                  BIT     #BIT3,@R5       ;RETRY?(BAD FLAG)
026466  001323                          BNE     64$             ;YES

026470  012737  000002  003342          MOV     #2,TEST9        ;INDICATE 2ND PART OF TST 8

026476  012700  027722          65$:    MOV     #TRBUF,R0       ;-->(POINT TO) XMIT BUFFER
026502  112710  000002                  MOVB    #RSCMND,@R0     ;FORM COMMAND MESSAGE PACK
026506  112760  000012  000001          MOVB    #RSMSIZ,1(R0)   ;THIS BIG
026514  112760  000001  000002          MOVB    #RSSNIT,2(R0)   ;OP CODE IS INITIALIZE
026522  013760  000064  000012          MOV     REC,10.(R0)     ;TO THIS RECORD
026530  105060  000003                  CLRB    3.(R0)          ;NO MODIFIER
026534  105060  000004                  CLRB    4.(R0)          ;NO DRIVE
026540  112760  000010  000005          MOVB    #BIT03,5.(R0)   ;SET MRSP SWITCH
026546  005060  000006                  CLR     6.(R0)          ;NO SEQUENCE #
026552  005060  000010                  CLR     8.(R0)          ;NO BYTE COUNT
026556  012701  000012                  MOV     #RSMSIZ,R1      ;GET COUNT
026562  005721                          TST     (R1)+           ;PLUS FLAG + BCNT
                                                                ;FOR CHECKSUM CALC
026564  004737  013654                  CALL    CHKSUM          ;R0-->TOP   R1=# OF BYTES
026570  010110                          MOV     R1,(R0)         ;INSERT INTO PACKET
```

```
                                                        ;SET UP EXPECTATIONS:
3874 026572  012765  000016  000070          MOV     #RSSNSZ,SNDCNT(R5)  ;HOW MANY TO SEND
3875 026600  112765  000002  000034          MOVB    #RSCMND,XSFLG(R5)   ;EXPECT END PACK
3876 026606  012765  000016  000036          MOV     #RSNDSZ,XSCNT(R5)   ;COUNT WITH THIS
3877 026614  012765  000001  000032          MOV     #1.,XSPKNM(R5)  ;EXPECT ONLY 1 PACKET
3878
3879 026622  004737  006560                  CALL    RSVP                ;SEND
3880                                                                    ;AND RETURN TO SCHEDULER
3881
3882 026626  032715  000010                  BIT     #BIT3,@R5           ;RETRY (FLAG BYTE ERROR)?
3883 026632  001321                          BNE     65$                 ;YES
3884
3885 026634  012700  027722          66$:    MOV     #TRBUF,R0           ;-->(POINT TO) XMIT BUFFER
3886 026640  112710  000002                  MOVB    #RSCMND,@R0         ;FORM COMMAND MESSAGE PACK
3887 026644  112760  000012  000001          MOVB    #RSMSIZ,1(R0)       ;THIS BIG
3888 026652  112760  000000  000002          MOVB    #RSSNOP,2(R0)       ;OP CODE IS NO-OPERATION
3889 026660  013760  000064  000012          MOV     REC,10.(R0)         ;TO THIS RECORD
3890 026666  105060  000003                  CLRB    3.(R0)              ;NO MODIFIER
3891 026672  105060  000004                  CLRB    4.(R0)              ;NO DRIVE
3892 026676  112760  000010  000005          MOVB    #BIT03,5.(R0)       ;SET MRSP SWITCH
3893 026704  005060  000006                  CLR     6.(R0)              ;NO SEQUENCE #
3894 026710  005060  000010                  CLR     8.(R0)              ;NO BYTE COUNT
3895 026714  012701  000012                  MOV     #RSMSIZ,R1          ;GET COUNT
3896 026720  005721                          TST     (R1)+               ;PLUS FLAG + BCNT
3897                                                                    ;FOR CHECKSUM CALC
3898 026722  004737  013654                  CALL    CHKSUM              ;R0-->TOP   R1=# OF BYTES
3899 026726  010110                          MOV     R1,(R0)             ;INSERT INTO PACKET
3900                                                                    ;SET UP EXPECTATIONS:
3901 026730  012765  000016  000070          MOV     #RSSNSZ,SNDCNT(R5)  ;HOW MANY TO SEND
3902 026736  112765  000002  000034          MOVB    #RSCMND,XSFLG(R5)   ;EXPECT END PACK
3903 026744  012765  000016  000036          MOV     #RSNDSZ,XSCNT(R5)   ;COUNT WITH THIS
3904 026752  012765  000001  000032          MOV     #1.,XSPKNM(R5)  ;EXPECT ONLY 1 PACKET
3905
3906 026760  004737  006560                  CALL    RSVP                ;SEND
3907                                                                    ;AND RETURN TO SCHEDULER
3908
3909 026764  032715  000010                  BIT     #BIT3,@R5           ;RETRY (FLAG BYTE ERROR)?
3910 026770  001321                          BNE     66$                 ;YES
3911
3912 026772  005237  003322                  INC     DONE
3913 026776  005037  003342                  CLR     TEST9
3914
3915 027002  005737  002224                  TST     PPSOT9              ;PROTOCOL SUMMARY @ END OF PASS
3916 027006  00147                           BEQ     ENDT9               ;NO
3917 027010  005037  027366                  CLR     UNITNO              ;SET UNIT # TO ZERO
3918 027014                                  PRINTF  #MSAGE1             ;PRINT HEADER
     027014  012746  027212                                                      MOV     #MSAGE1,-(SP)
     027020  012746  000001                                                      MOV     #1,-(SP)
     027024  010600                                                              MOV     SP,R0
     027026  104417                                                              TRAP    C$PNTF
     027030  062706  000004                                                      ADD     #4,SP
3919 027034  012737  003350  003312          MOV     #BLKTBL,DEVPTR      ;SET ALL UNITS
3920 027042  017705  154244          1$:     MOV     @DEVPTR,R5          ;GET POINTER
3921 027046  005765  000000                  TST     STATUS(R5)          ;IS UNIT ABORTED
3922 027052  100431                          BMI     3$                  ;YES
3923 027054  005765  000210                  TST     MRSP(R5)            ;IS UNIT MODIFIED
3924 027060  001413                          BEQ     2$                  ;NO
```

```
                                                PRINTF  #MSAGE2,UNITNO  ;MESSAGE FOR MODIFIED UNIT
                                                                                MOV     UNITNO,-(SP)
                                                                                MOV     #MSAGE2,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #6,SP
                                                BR      4$              ;SEE IF LAST UNIT
                                        2$:     PRINTF  #MSAGE3,UNITNO  ;MESSAGE FOR NON-MODIFIED UNIT
                                                                                MOV     UNITNO,-(SP)
                                                                                MOV     #MSAGE3,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #6,SP
                                                BR      4$              ;SEE IF LAST UNIT
                                        3$:     PRINTF  #MSAGE4,UNITNO  ;MESSAGE FOR ABORTED UNIT
                                                                                MOV     UNITNO,-(SP)
                                                                                MOV     #MSAGE4,-(SP)
                                                                                MOV     #2,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #6,SP
027162  023727  003312  003366  4$:     CMP     DEVPTR,#LSTDEV  ;IS THIS THE LAST DEVICE
027170  103006                          BHIS    ENDT9           ;YES
027172  062737  000002  003312          ADD     #2,DEVPTR       ;GET NEXT UNIT
027200  005237  027366                  INC     UNITNO          ;INC UNIT #
027204  000716                          BR      1$

027206  000207                  ENDT9:  RETURN

                                        ENDTST
027210                                                                  L10024:
027210                                                                          TRAP    C$ETST
027210  104401

027212      045     116     045 MSAGE1: .ASCIZ  /%N%S8%AUNIT NO%S9%S6%APROTOCOL%N/
027253      045     116     045 MSAGE2: .ASCIZ  !%N%S9%S2%O1%S9%S9%ARSP/MRSP!
027307      045     116     045 MSAGE3: .ASCIZ  /%N%S9%S2%O1%S9%S9%ARSP/
027336      045     116     045 MSAGE4: .ASCIZ  /%N%S9%S2%O1%S9%S9%A---/
                                        .EVEN
027366  000000                  UNITNO: .WORD
```

```
                                .SBTTL  PATCH AREA
000144                          .REPT   100.
                                .WORD
                                .ENDR
```

```
                                .SBTTL  I/O BUFFER AREAS:

                                ;WHO-GETS-WHAT-SPACE TABLE

027700  030760                BUFTBL: .WORD   BUF0
027702  032016                        .WORD   BUF1
027704  033054                        .WORD   BUF2
027706  034112                        .WORD   BUF3
027710  035150                        .WORD   BUF4
027712  036206                        .WORD   BUF5
027714  037244                        .WORD   BUF6
027716  040302                        .WORD   BUF7

                                ;-----------------------------------------------------------------
                                ;ONLY 1 TRANSMIT BUFFER NECESSARY:

027720      023                             .BYTE RSXOFF
027721      023                             .BYTE RSXOFF    ;SEND XOFF BEFORE EVERY PACKET

027722                                      TRBUF:  .BLKB RCBFSZ
                                ;-----------------------------------------------------------------

030760                          BUF0:   .BLKB   RCBFSZ
032016                          BUF1:   .BLKB   RCBFSZ
033054                          BUF2:   .BLKB   RCBFSZ
034112                          BUF3:   .BLKB   RCBFSZ
035150                          BUF4:   .BLKB   RCBFSZ
036206                          BUF5:   .BLKB   RCBFSZ
037244                          BUF6:   .BLKB   RCBFSZ
040302                          BUF7:   .BLKB   RCBFSZ

                                ;-----------------------------------------------------------------

041340                                  ENDMOD
```

```
                                .TITLE PARAMETER CODING

                                .SBTTL  HARDWARE PARAMETER CODING SECTION

041340                                  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.
                                ;--

041340                                  BGNHRD
041340  000021                                                          .WORD L10025-L$HARD/2
041342                                                          L$HARD::

041342                                  GPRMA   MSG1,0,0,160000,177777,YES
041342  000031                                                          .WORD   T$CODE
041344  041404                                                          .WORD   MSG1
041346  160000                                                          .WORD   T$LOLIM
041350  177777                                                          .WORD   T$HILIM
041352                                  GPRMA   MSG1B,2,0,0,776,YES
041352  001031                                                          .WORD   T$CODE
041354  041415                                                          .WORD   MSG1B
041356  000000                                                          .WORD   T$LOLIM
041360  000776                                                          .WORD   T$HILIM
041362                                  GPRML   MSG1C,6,1,YES
041362  003130                                                          .WORD   T$CODF
041364  041432                                                          .WORD   MSG1C
041366  000001                                                          .WORD   1
041370                                  GPRML   MSG2,4,1,YES
041370  002130                                                          .WORD   T$CODE
041372  041450                                                          .WORD   MSG2
041374  000001                                                          .WORD   1
041376                                  GPRML   MSG3,4,2,YES
041376  002130                                                          .WORD   T$CODE
041400  041465                                                          .WORD   MSG3
041402  000002                                                          .WORD   2

041404                                  ENDHRD
                                                                .EVEN
041404                                                          L10025:

041404  124     125     065     MSG1:   .ASCIZ  /TU58 CSR/
041415  126     105     103     MSG1R:  .ASCIZ  /VECTOR ADDR./
041432  120     104     124     MSG1C:  .ASCIZ  /PDT INTERFACE/
041450  124     105     123     MSG2:   .ASCIZ  /TEST DRIVE 0/
041465  124     105     123     MSG3:   .ASCIZ  /TEST DRIVE 1/
                                        .EVEN
```

```
                                .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.
                                ;--

                                        BGNSFT
                                                                                .WORD L10026-L$SOFT/2
041502  00003                                                           L$SOFT::
041504
                                        GPRMD   MSG4,0,D,1777,8.,512.,YES
                                                                                .WORD   T$CODE
041504  000052                                                                  .WORD   MSG4
041506  041566                                                                  .WORD   1777
041510  001777                                                                  .WORD   T$LOLIM
041512  000010                                                                  .WORD   T$HILIM
041514  001000
                                        GPRML   MSG4B,10,1,YES
                                                                                .WORD   T$CODE
041516  004130                                                                  .WORD   MSG4B
041520  041633                                                                  .WORD   1
041522  000001
                                        GPRML   MSG5,2,1,YES
                                                                                .WORD   T$CODE
041524  001130                                                                  .WORD   MSG5
041526  041675                                                                  .WORD   1
041530  000001
                                        GPRML   MSG6,6,1,YES
                                                                                .WORD   T$CODE
041532  003130                                                                  .WORD   MSG6
041534  041727                                                                  .WORD   1
041536  000001
                                        GPRML   MSG7,4,1,YES
                                                                                .WORD   T$CODE
041540  002130                                                                  .WORD   MSG7
041542  041754                                                                  .WORD   1
041544  000001
                                        GPRMD   MSG8,10.,D,377,1,254.,YES
                                                                                .WORD   T$CODE
041546  005052                                                                  .WORD   MSG8
041550  042002                                                                  .WORD   377
041552  000377                                                                  .WORD   T$LOLIM
041554  000001                                                                  .WORD   T$HILIM
041556  000376
                                        GPRML   MSG9,12.,1,YES
                                                                                .WORD   T$CODE
041560  006130                                                                  .WORD   MSG9
041562  042043                                                                  .WORD   1
041564  000001
                                SFTOUT: ENDSFT
                                                                                .EVEN
041566                                                                  L10026:
                                MSG4:   .ASCIZ  'NUMBER OF BLOCKS:TEST 5-8 (8 TO 512)'
                                MSG4B:  .ASCIZ  /ADD DR # TO DATA PATTERN:TEST 5-8/
                                MSG5:   .ASCIZ  /STATISTICS PRINTED AT EOP/
                                MSG6:   .ASCIZ  /COMPARE DATA ON READ/
                                MSG7:   .ASCIZ  /PRINT PACKET ON ERROR/
                                MSG8:   .ASCIZ  /# ERRORS = DVC FATAL IF 'EVL'SET/
                                MSG9:   .ASCIZ  /PRINT UNIT PROTOCOL SUMMARY (TEST 9)/
                                        .EVEN
```

```
4133          000016                      .REPT   14.             ;LASTAD CORRECTION
4134                                      .WORD
4135                                      .ENDR
4142 042144                               LASTAD
                                                                          .EVEN
      042144  042164                                                     .WORD T$FREE
      042146  000006                                                     .WORD T$SIZE
      042150                        L$LAST::
4143 042150                               ENDMOD
4144
4145 042150                               BGNSETUP        1
4146 042150                               BGNPTAB
      042150  000000                                                     .WORD   0
      042152  000004                                                     .WORD   L10031-./2-1
      042154                                                    L10027:
4147 042154  176500                       176500
4148 042156  000300                       300
4149 042160  000003                       3
4150 042162  000000                       0
4151 042164                               ENDPTAB
      042164                                                    L10031:
4152 042164                               ENDSETUP
4153          000001                .END
```

PARAMETER CODING MACRO M1113 25-FEB-82 07:06 PAGE 127-1 SEQ 0125

SYMBOL TABLE

```
ABNDX = 000004 G    CHKANS  010576 G    C$MSG = 000023      ESSLF = 177777 G    G$OFSI= 000376
ABO     013146      CHKEND  011302 G    C$OPEN= 000034      ESTRY = 000001 G    G$PRMA= 000001
ABOMSG  017254      CHKERR  011406      C$PNTB= 000014      ESWLOC= 177765 G    G$PRMD= 000002
ABONM   006532      CHKPKS  010706 G    C$PNTF= 000017      ESWR  = 177757      G$PRML= 000000
ADR   = 000020 G    CHKPTR  010704      C$PNTS= 000016      EVL   = 000004 G    G$RADA= 000140
ALLGON  003340 G    CHKREE  011460      C$PNTX= 000015      EVLTHR  002222      G$RADB= 000000
ASSEMB= 000010      CHKRET  012024      C$QIO = 000377      EXOFF   007427      G$RADD= 000040
BDATA = 000134 G    CHKSUC  012320 G    C$RDBU= 000007      EXON    007426      G$RADL= 000120
BDBYTS  014726      CHKSUM  013654 G    C$REFG= 000047      E$END = 002100      G$RADO= 000020
BDCHK = 000022 G    CHK8    010616      C$RESE= 000033      E$LOAD= 000035      G$XFER= 000004
BDCOM = 000014 G    CKCKSM  013750 G    C$REVI= 000003      FLGLOC  016752 G    G$YES = 000010
BIT0  = 000001 G    CLRALL  005710 G    C$RFLA= 000021      FM      015730      HARDR = 000136 G
BIT00 = 000001 G    CLRBUF  005750 G    C$RPT = 000025      FM0     015712      HARDW = 000140 G
BIT01 = 000002 G    CLRPTR  006002      C$SEFG= 000046      FTLNM   003520      HELP  = 000000
BIT02 = 000004 G    CMDSNT= 000100 G    C$SPRI= 000041      F$AU  = 000015      HOE   = 100000 G
BIT03 = 000010 G    CMNDER= 000040 G    C$SVEC= 000037      F$AUTO= 000020      HRD     013126
BIT04 = 000020 G    CMPDAT  002216      C$TPRI= 000013      F$BGN = 000040      HRDRD = 000016 G
BIT05 = 000040 G    CNINIT= 000032 G    DESC    014730      F$CLEA= 000007      HRDWR = 000020 G
BIT06 = 000100 G    COMPAR  014550 G    DEVPTR  003312 G    F$DU  = 000016      HRD1    011746
BIT07 = 000200 G    CSNRDY  003344 G    DEV0    003370      F$END = 000041      IBE   = 010000 G
BIT08 = 000400 G    CSRCVB  003346 G    DEV1    003602      F$HARD= 000004      IDPTR   003324 G
BIT09 = 001000 G    C$AU  = 000052      DEV2    004014      F$HW  = 000013      IDU   = 000040 G
BIT1  = 000002 G    C$AUTO= 000061      DEV3    004226      F$INIT= 000006      IER   = 020000 G
BIT10 = 002000 G    C$BRK = 000022      DEV4    004440      F$JMP = 000050      INIT    016164
BIT11 = 004000 G    C$BSEG= 000004      DEV5    004652      F$MOD = 000000      INITWD  014542
BIT12 = 010000 G    C$BSUB= 000002      DEV6    005064      F$MSG = 000011      INIT2   016212
BIT13 = 020000 G    C$CEFG= 000045      DEV7    005276      F$PROT= 000021      ISR   = 000100 G
BIT14 = 040000 G    C$CLCK= 000062      DFPTBL  002176 G    F$PWR = 000017      IXE   = 004000 G
BIT15 = 100000 G    C$CLEA= 000012      DFTL1   012774      F$RPT = 000012      I$AU  = 000041
BIT2  = 000004 G    C$CLOS= 000035      DIAGMC= 000000      F$SEG = 000003      I$AUTO= 000041
BIT3  = 000010 G    C$CLP1= 000006      DLV   = 000074 G    F$SOFT= 000005      I$CLN = 000041
BIT4  = 000020 G    C$CVEC= 000036      DOBRK   014010 G    F$SRV = 000010      I$DU  = 000041
BIT5  = 000040 G    C$DCLN= 000044      DONE    003322 G    F$SUB = 000002      I$HRD = 000041
BIT6  = 000100 G    C$DODU= 000051      DR    = 000060 G    F$SW  = 000014      I$INIT= 000041
BIT7  = 000200 G    C$DRPT= 000024      DRVCHK  002220      F$TEST= 000001      I$MOD = 000041
BIT8  = 000400 G    C$DU  = 000053      EF.CON= 000036 G    GBTMP   010572      I$MSG = 000041
BIT9  = 001000 G    C$EDIT= 000003      EF.NEW= 000035 G    GBTMP2  010574      I$PROT= 000040
BLKEND= 000202 G    C$ERDF= 000055      EF.PWR= 000034 G    GETANS  007152 G    I$PTAB= 000041
BLKER   003332 G    C$ERHR= 000056      EF.RES= 000037 G    GETHRD  016316      I$PWR = 000041
BLKSIZ= 000212 G    C$ERRO= 000060      EF.STA= 000040 G    GETPTR  007216      I$RPT = 000041
BLKTBL  003350 G    C$ERSF= 000054      ENDGP8  010146      GETR5   017222      I$SEG = 000041
BOE   = 000400 G    C$ERSO= 000057      ENDRSP  007100      GTAGIN  007464      I$SETU= 000041
BRKPTR  014546      C$ESCA= 000010      ENDT9   027206      GTBYTE  010346 G    I$SFT = 000041
BRKTO   014544      C$ESEG= 000005      ERRDES  013170 G    GTDOWN  010104      I$SRV = 000041
BRKWD   014540      C$ESUB= 000003      ERRMOD  010150      GTOK    007730      I$SUB = 000041
BUFTBL  027700      C$ETST= 000001      ESABO = 177720 G    GTPKS1  007220 G    I$TST = 000041
BUF0    030760      C$EXIT= 000032      ESCKS = 177757 G    GTPKS8  007430 G    J$JMP = 000167
BUF1    032016      C$GETB= 000026      ESCKSM= 177757      GTPTR   010344      LENGTH  002210
BUF2    033054      C$GETW= 000027      ESCMD = 177720 G    GTUM    007672      LGOFST= 000120 G
BUF3    034112      C$GMAN= 000043      ESNCRT= 177767 G    G$CNTO= 000200      LNCNT   015130
BUF4    035150      C$GPHR= 000042      ESNOMO= 177737 G    G$DELM= 000372      LOE   = 040000 G
BUF5    036206      C$GPLO= 000030      ESNONX= 177770 G    G$DISP= 000003      LOG     012634 G
BUF6    037244      C$GPRI= 000040      ESOK  = 000000 G    G$EXCP= 000400      LOGO    013156
BUF7    040302      C$INIT= 000011      ESPART= 177776 G    G$HILI= 000002      LOGOK   012722
CARLF   015144      C$INLP= 000020      ESRD  = 177757      G$LOLI= 000001      LOGOK2  013006
CHECK   016274      C$MANI= 000050      ESREC = 177711 G    G$NO  = 000000      LOGO    012710
CHKANR  010702      C$MEM = 000031      ESSK  = 177740 G    G$OFFS= 000400      LOG1    013024
```

SYMBOL TABLE

```
LOG2     013054          L$UNIT   002012 G    MSNRSP   002764 G    PRIO7  = 000340 G    SETUP    006004 G
LOG3     013114          L10001   002206      MSOVRN   003260 G    PRNPAK   014764 G    SFPTBL   002210 G
LOG3B    013136          L10002   002226      MSPART   002634 G    PRNSIZ   003336 G    SFT      013104
LOT    = 000010 G        L10003   013360      MSQRSP   003000 G    PTR      017252      SFTOUT   041566
LSTDEV   003366 G        L10004   014446      MSREC    002720 G    RCBCNT   003316      SFTRD  = 000002 G
L$ACP    002110 G        L10005   014502      MSRNIT   002536 G    RCBFSZ=  001036 G    SFTWR  = 000004 G
L$APT    002036 G        L10006   015572      MSSELF   002364 G    RCDB   = 000024 G    SKERR  = 000024 G
L$AU     017302 G        L10007   016662      MSSFRD   003054 G    RCFLG    003314 G    SLFER  = 000044 G
L$AUT    002070 G        L10010   017060      MSSFWR   003114 G    RCINIT=  000006 G    SND      006650
L$AUTO   016754 G        L10011   017154      MSSKER   002306 G    RCSR   = 000022 G    SNDBYT   007102 G
L$CCP    002106 G        L10012   017220      MSTOSN   003032 G    RCVBUF=  000102 G    SNDCNT=  000070 G
L$CLEA   017136 G        L10013   017302      MSUNIT   002656 G    RCVHND   014450      SNDHND   014434
L$CO     002032 G        L10014   017504      MSWPRO   002514 G    RCVINT   014450 G    SNDINT   014434 G
L$DEPO   002011 G        L10015   017756      MSWRSP   002740 G    RDN0   = 000114 G    SOFTR  = 000122 G
L$DESC   002122 G        L10016   021350      MXRTRY   003330 G    RDN1   = 000116 G    SOFTW  = 000124 G
L$DESP   002076 G        L10017   022754      NCART  = 000054 G    REC    = 000064 G    SRVTBL   010322
L$DEVP   002060 G        L10020   023744      NODRVS   016714      RECDAT   017742      STAEOP   002212
L$DISP   002152 G        L10021   024530      NOMOR    006534      RECERR=  000042 G    STATHD   015600
L$DLY    002116 G        L10022   025520      NOMOT  = 000030 G    RECID    013442 G    STATUS=  000000 G
L$DTP    002040 G        L10023   026304      NOREE    011334      RECID2   013616      STHD2    016054
L$DTYP   002034 G        L10024   027210      NOUNIT=  000036 G    RECOV    012044      STRT     016746 G
L$DU     017156 G        L10025   041404      NOXOFF   006632      RETERR   012246      SUCCS  = 000076 G
L$DUT    002072 G        L10026   041566      NTSFT    013036      RETRY  = 000002 G    SUCOTL=  000046 G
L$DVTY   005510 G        L10027   042154      NXTRET   006530      RLUN     015574      SVCGBL=  000000
L$EF     002052 G        L10031   042164      NXTST    006102 G    RPTR     015576      SVCINS=  000001
L$ENVI   002044 G        MABEE    013066      NXTST2   006352      RSCMND=  000002 G    SVCSUB=  000001
L$ETP    002102 G        MESMRS   010200      ONEFIL=  000001      RSCONT=  000020 G    SVCTAG=  000001
L$EXP1   002046 G        MODRSP   010252      OTL    = 000010 G    RSDASZ=  000204 G    SVCTST=  000001
L$EXP4   002064 G        MRSDLY   010254      OVRFLO   013534      RSDATA=  000001 G    SWAPDR   005530 G
L$EXP5   002066 G        MRSP   = 000210 G    OVRN   = 000012 G    RSDNSZ=  000222 G    SWPTR    005630
L$HARD   041342 G        MSAGE1   027212      O$APTS=  000000      RSEND  = 000002 G    SYSTAT   003306 G
L$HIME   002120 G        MSAGE2   027253      O$AU   = 000001      RSGCDP=  000034 G    S$LSYM=  010000
L$HPCP   002016 G        MSAGE3   027307      O$BGNR=  000001      RSINIT=  000004 G    TAPLEN   003310 G
L$HPTP   002022 G        MSAGE4   027336      O$BGNS=  000001      RSMSIZ=  000012 G    TEST9    003342 G
L$HW     002176 G        MSAUTO   017116      O$DU   = 000001      RSNDSZ=  000016 G    THRSHI   012152
L$ICP    002104 G        MSBDA    002340 G    O$ERRT=  000000      RSNTAB   002226      THRSLO   012124
L$INIT   016164 G        MSCMD    002704 G    O$GNSW=  000001      RSSEND=  000100 G    TMP    = 000066 G
L$LADP   002026 G        MSCOM    002404 G    O$POIN=  000001      RSSGET=  000012 G    TOMANY   016664
L$LAST   042150 G        MSG1     041404      O$SETU=  000001      RSSNIT=  000001 G    TORCVB=  000050 G
L$LOAD   002100 G        MSG1B    041415      PARTL  = 000034 G    RSSNOP=  000000 G    TOSNDB=  000056 G
L$LUN    002074 G        MSG1C    041432      PATTEN=  000072 G    RSSNSZ=  000016 G    TRBUF    027722
L$MREV   002050 G        MSG2     041450      PDTFLG   016750 G    RSSRD  = 000002 G    TRK    = 000062 G
L$NAME   002000 G        MSG3     041465      PERDEV   006364      RSSSEK=  000005 G    TRPHND   017064
L$PRIO   002042 G        MSG4     041566      PKPTR  = 000104 G    RSSSLF=  000007 G    TRPPTR   017062
L$PROT   002142 G        MSG4B    041633      PNT    = 001000 G    RSSWR  = 000003 G    TSTPC  = 000020 G
L$PRT    002112 G        MSG5     041675      PPSOT9   002224      RSVP     006560 G    TSTTOP   003326
L$REPP   002062 G        MSG6     041727      PRBUF    002214      RSXOFF=  000023 G    TST1     017350
L$REV    002010 G        MSG7     041754      PRDAT    015132      RSXON  = 000020 G    TST2     017552
L$RPT    015150 G        MSG8     042002      PRFORM   015134      RTRYN    012204      TST3     020024
L$SOFT   041504 G        MSG9     042043      PRI    = 002000 G    RUN      006052 G    TST3PT   022742
L$SPC    002056 G        MSHCHK   002556 G    PRI00  = 000000 G    SAVCNT=  000206 G    TST4     021416
L$SPCP   002020 G        MSHDRD   003154 G    PRI01  = 000040 G    SECREC   003334 G    TST5     023022
L$SPTP   002024 G        MSHDWR   003216 G    PRI02  = 000100 G    SERVST   010342      TST5EX   023736
L$STA    002030 G        MSNIT    002620 G    PRI03  = 000140 G    SETDR    005632 G    TST6     024012
L$SW     002210 G        MSNLOG   002322 G    PRI04  = 000200 G    SETLEN   016624      TST6EX   024522
L$TEST   002114 G        MSNOMO   002446 G    PRI05  = 000240 G    SETPTR   005706      TST7     024576
L$TIML   002014 G        MSNOTP   002464 G    PRI06  = 000300 G    SETSRV   010256      TST7EX   025512
```

SYMBOL TABLE

```
TST8    025566          T$NS0 = 000000        T$$AUT= 010010        T1TRY = 000146 G      WAIT    014504
TST8EX  026276          T$NS1 = 000005        T$$CLE= 010011        T2      017506 G      WHCHDR  013640 G
TST9    026330          T$PCNT= 000000        T$$DAT= 010031        T3      017760 G      WRLOCK= 000026 G
TUVECT= 000204 G        T$PTAB= 010030        T$$DU = 010012        T4      021352 G      WRTN0 = 000110 G
T$ARGC= 000002          T$PTHV= 000001        T$$HAR= 010025        T4TRY = 000132 G      WRTN1 = 000112 G
T$CODE= 006130          T$PTNU= 000001        T$$HW = 010001        T5      022756 G      XFNSND  006640
T$ERRN= 000146          T$SAVL= 177777        T$$INI= 010007        T6      023746 G      XMDB  = 000030 G
T$EXCP= 000000          T$SEGL= 177777        T$$MSG= 010003        T7      024532 G      XMSR  = 000026 G
T$FLAG= 000040          T$SIZE= 000006        T$$PC = 000001        T8      025522 G      XSCNT = 000036 G
T$FREE= 042164          T$SUBN= 000000        T$$PRO= 010000        T9      026306 G      XSFLG = 000034 G
T$GMAN= 000000          T$TAGL= 177777        T$$PTA= 010030        UAM   = 000200 G      XSPKNM= 000032 G
T$HILI= 000376          T$TAGN= 010032        T$$RPT= 010006        UNIT    013362 G      XSPTR = 000106 G
T$LAST= 000001          T$TEMP= 000000        T$$SOF= 010026        UNITNO  027366        X$ALWA= 000000
T$LOLI= 000001          T$TEST= 000011        T$$SRV= 010005        UNREC   012224        X$FALS= 000040
T$LSYM= 010000          T$TSTM= 177777        T$$SW = 010002        UNSUC   011622        X$OFFS= 000400
T$LTNO= 000011          T$TSTS= 000001        T$$TES= 010024        UNXPCT  007634        X$TRUE= 000020
T$NEST= 177777          T$$AU = 010013        T1      017304 G

. ABS.  042164     000
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  33938 WORDS  ( 133 PAGES)
DYNAMIC MEMORY:  21558 WORDS  ( 82 PAGES)
ELAPSED TIME:  00:05:18
CZTUUDO.BIN/EN:AMA:ABS,CZTUUDO/CR/-SP=LB1:[1,1]SVC/MLB,SY:[203,377]CZTUUDO
```

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08 PAGE 1 SEQ 0128
SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE		REFERENCES								
ABNDX	= 000004	G	#16-901	*72-2691	72-2704	72-2706					
ABO	013146		72-2701	72-2715	72-2728	#72-2744					
ABOMSG	017254		97-3448	#97-3478							
ABONM	006532		*49-1812	*49-1829	49-1834	#49-1842					
ADR	= 000020	G	#10-653								
ALLGON	003340	G	#15-870	*49-1838	*91-3313	95-3405					
ASSEMB	= 000010		5-464	5-464							
BDATA	= 000134	G	#17-953	89-3185	89-3195						
BDBYTS	014726		*86-3033	*86-3044	86-3047	86-3052	#86-3062				
BDCHK	= 000022	G	#11-681	66-2397	66-2454						
BDCOM	= 000014	G	#11-678	68-2504	72-2716						
BIT0	= 000001	G	#10-653	37-1628	60-2225	72-2707	78-2818	78-2819	78-2821	78-2832	82-2907
BIT00	= 000001	G	#10-653	10-653							
BIT01	= 000002	G	#10-653	10-653							
BIT02	= 000004	G	#10-653	10-653							
BIT03	= 000010	G	#10-653	10-653	117-3865	117-3892					
BIT04	= 000020	G	#10-653	10-653							
BIT05	= 000040	G	#10-653	10-653							
BIT06	= 000100	G	#10-653	10-653							
BIT07	= 000200	G	#10-653	10-653							
BIT08	= 000400	G	#10-653	10-653							
BIT09	= 001000	G	#10-653	10-653							
BIT1	= 000002	G	#10-653	49-1794	54-1985	60-2226	64-2326	68-2490	68-2561		
BIT10	= 002000	G	#10-653	49-1792	50-1914	68-2510	68-2530	68-2538	68-2559	106-3631	106-3631
			106-3632	107-3666	107-3666	107-3667	109-3709	109-3709	111-3740	113-3771	113-3771
			115-3802								
BIT11	= 004000	G	#10-653	89-3171	91-3250	91-3271	93-3372				
BIT12	= 010000	G	#10-653	106-3631	106-3631	106-3631	107-3666	107-3666	107-3666	109-3709	109-3709
			109-3709	113-3771	113-3771	113-3771					
BIT13	= 020000	G	#10-653	49-1782	49-1790	66-2413	66-2425	68-2494			
BIT14	= 040000	G	#10-653	49-1818	49-1821	70-2631	82-2957	91-3277			
BIT15	= 100000	G	#10-653	37-1626	49-1801	56-2027	56-2049	58-2092	58-2111	58-2131	58-2162
			64-2334	68-2496	91-3249	91-3296	93-3372	97-3445			
BIT2	= 000004	G	#10-653	49-1789	49-1803	54-1985	60-2227	64-2326	68-2490	68-2492	68-2561
BIT3	= 000010	G	#10-653	49-1780	60-2228	66-2377	66-2413	66-2425	102-3582	104-3600	106-3631
			106-3631	106-3631	106-3632	107-3666	107-3666	107-3666	107-3667	109-3709	109-3709
			109-3709	111-3740	113-3771	113-3771	113-3771	115-3802	117-3853	117-3882	117-3909
BIT4	= 000020	G	#10-653	50-1908	50-1912	60-2229	74-2763	82-2909			
BIT5	= 000040	G	#10-653	50-1930	50-1932	60-2230					
BIT6	= 000100	G	#10-653	60-2231	68-2501	68-2562	82-2934	82-2942	82-2949	82-2965	82-2966
			84-2978	84-2986	86-3053						
BIT7	= 000200	G	#10-653	60-2232	68-2530	68-2538	68-2544	68-2546	68-2548	68-2559	107-3666
			107-3667								
BIT8	= 000400	G	#10-653	39-1661	49-1784	50-1918	50-1927	68-2523	68-2552	91-3278	
BIT9	= 001000	G	#10-653	37-1630	50-1902	68-2507	68-2534	70-2617	91-3280		
BLKEND	= 000202	G	#17-973	91-3288							
BLKER	003332	G	#15-867	*50-1899	*91-3321						
BLKSIZ	= 000212	G	#17-977	19-999	19-1000	19-1001	19-1002	19-1003	19-1004	19-1005	19-1006
BLKTBL	003350	G	#19-987	37-1624	39-1658	41-1683	45-1726	49-1776	49-1813	58-2090	64-2332
			89-3165	91-3245	91-3265	93-3370	97-3469	117-3919			
BOE	= 000400	G	#10-653								
BRKPTR	014546		*82-2933	*82-2941	*82-2948	84-2999	#84-3013				

SYMBOL CROSS REFERENCE
CREF

SYMBOL	VALUE		REFERENCES								
BRKTO	014544		*82-2906	*82-2915	*82-2921	*82-2931	*82-2939	*84-3003	#84-3012		
BRKWD	014540		82-2920	#84-3009							
BUFTBL	027700		91-3252	#121-3958							
BUF0	030760		121-3958	#121-3978							
BUF1	032016		121-3959	#121-3979							
BUF2	033054		121-3960	#121-3980							
BUF3	034112		121-3961	#121-3981							
BUF4	035150		121-3962	#121-3982							
BUF5	036206		121-3963	#121-3983							
BUF6	037244		121-3964	#121-3984							
BUF7	040302		121-3965	#121-3985							
CARLF	015144		88-3091	88-3093	#88-3102						
CHECK	016274		91-3255	#91-3260							
CHKANR	010702		64-2330	64-2342	#64-2346						
CHKANS	010576	G	47-1751	#64-2324							
CHKEND	011302	G	66-2402	66-2450	#68-2487						
CHKERR	011406		68-2500	#68-2507							
CHKPKS	010706	G	64-2328	64-2340	#66-2376						
CHKPTR	010704		*64-2332	64-2333	64-2341	*64-2343	#64-2348				
CHKREE	011460		68-2493	#68-2514							
CHKRET	012024		68-2498	68-2503	68-2506	68-2509	68-2513	68-2531	68-2539	68-2547	68-2549
			#68-2561								
CHKSUC	012320	G	68-2495	68-2514	#70-2596						
CHKSUM	013654	G	#78-2816	80-2859	102-3582	104-3600	106-3631	106-3631	106-3631	106-3632	106-3632
			107-3666	107-3666	107-3666	107-3667	107-3667	109-3709	109-3709	109-3709	111-3740
			111-3740	113-3771	113-3771	113-3771	115-3802	115-3802	117-3843	117-3871	117-3898
CHK8	010616		64-2327	#64-2332							
CKCKSM	013750	G	66-2395	66-2433	66-2445	#80-2856					
CLRALL	005710	G	#41-1683	54-1988							
CLRBUF	005750	G	41-1685	#43-1701	54-1991						
CLRPTR	006002		*41-1683	41-1684	41-1686	*41-1688	#43-1711				
CMDSNT	= 000100	G	#16-926	*50-1911	70-2603	70-2607	70-2620	74-2769	*82-2908		
CMNDER	= 000040	G	#11-688	70-2641							
CMPDAT	002216		#9-579	86-3035							
CNINIT	= 000032	G	#11-685	82-2961							
COMPAR	014550	G	66-2411	66-2435	#86-3030						
CSNRDY	003344	G	#15-877	52-1956							
CSRCVB	003346	G	#15-878	62-2262							
C$AU	= 000052		#5-464	99-3513							
C$AUTO	= 000061		#5-464	93-3381							
C$BRK	= 000022		#5-464	47-1749	49-1837	52-1960	62-2284	82-2911	84-3001	84-3002	89-3164
			89-3167	89-3169							
C$BSEG	= 000004		#5-464								
C$BSUB	= 000002		#5-464								
C$CEFG	= 000045		#5-464								
C$CLCK	= 000062		#5-464								
C$CLEA	= 000012		#5-464	95-3430							
C$CLOS	= 000035		#5-464								
C$CLP1	= 000006		#5-464								
C$CVEC	= 000036		#5-464	82-2967	82-2969	93-3380					
C$DCLN	= 000044		#5-464	49-1839	91-3263	91-3283					
C$DODU	= 000051		#5-464	72-2746	93-3392						

SYMBOL CROSS REFERENCE
CREF

SYMBOL	VALUE		REFERENCES								
C$DRPT	= 000024		#5-464	95-3409							
C$DU	= 000053		#5-464	97-3468							
C$EDIT	= 000003		#5-464	5-507							
C$ERDF	= 000055		#5-464	72-2700	72-2714						
C$ERHR	= 000056		#5-464	72-2727	72-2742						
C$ERRO	= 000060		#5-464								
C$ERSF	= 000054		#5-464	49-1836	91-3262	91-3282					
C$ERSO	= 000057		#5-464	72-2720	72-2738	86-3051					
C$ESCA	= 000010		#5-464								
C$ESEG	= 000005		#5-464								
C$ESUB	= 000003		#5-464								
C$ETST	= 000001		#5-464	102-3587	104-3614	106-3649	107-3691	109-3723	111-3754	113-3785	115-3816
			117-3938								
C$EXIT	= 000032		#5-464	102-3580	104-3594	106-3624	107-3659	109-3699	111-3730	113-3761	1.5-3792
			117-3828								
C$GETB	= 000026		#5-464								
C$GETW	= 000027		#5-464								
C$GMAN	= 000043		#5-464								
C$GPHR	= 000042		#5-464	91-3269							
C$GPLO	= 000030		#5-464								
C$GPRI	= 000040		#5-464								
C$INIT	= 000011		#5-464	91-3350							
C$INLP	= 000020		#5-464								
C$MANI	= 000050		#5-464								
C$MEM	= 000031		#5-464								
C$MSG	= 000023		#5-464	74-2776							
C$OPEN	= 000034		#5-464								
C$PNTB	= 000014		#5-464	68-2536	74-2766	74-2769	74-2772	86-3052			
C$PNTF	= 000017		#5-464	58-2191	88-3086	88-3091	88-3093	93-3389	97-3448	117-3918	117-3925
			117-3927	117-3929							
C$PNTS	= 000016		#5-464	89-3166	89-3168	89-3184	89-3185	89-3195			
C$PNTX	= 000015		#5-464	68-2512	68-2517	68-2520	68-2522	68-2543	68-2551		
C$QIO	= 000377		#5-464								
C$RDBU	= 000007		#5-464								
C$REFG	= 000047		#5-464	91-3242							
C$RESE	= 000033		#5-464	#5-464							
C$REVI	= 000003		#5-464	5-507							
C$RFLA	= 000021		#5-464	91-3320							
C$RPT	= 000025		#5-464	89-3208							
C$SEFG	= 000046		#5-464								
C$SPRI	= 000041		#5-464	82-2926							
C$SVEC	= 000037		#5-464	82-2927	82-2929	93-3369					
C$TPRI	= 000013		#5-464								
DESC	014730		86-3052	#86-3063							
DEVPTR	003312	G	#15-859	*49-1776	49-1777	49-1807	*49-1809	*49-1813	49-1814	49-1830	*49-1832
			*91-3245	91-3247	91-3254	*91-3256	*91-3265	91-3267	*91-3314	*117-3919	117-3920
			117-3930	*117-3932							
DEV0	003370		19-987	#19-999							
DEV1	003602		19-988	#19-1000							
DEV2	004014		19-989	#19-1001							
DEV3	004226		19-990	#19-1002							
DEV4	004440		19-991	#19-1003							

SYMBOL CROSS REFERENCE
CREF

SYMBOL	VALUE		REFERENCES								
DEV5	004652		19-992	#19-1004							
DEV6	005064		19-993	#19-1005							
DEV7	005276		19-994	#19-1006							
DFPTBL	002176	C	#8-552								
DFTL1	012774		*72-2712	*72-2713	#72-2714	72-2714					
DIAGMC	= 000000		5-464	5-464							
DLV	= 000074	G	#16-924	56-2047	58-2133	58-2164	*62-2265	62-2266	62-2267	*62-2269	*62-2290
			62-2291	62-2292	*62-2294	66-2415	74-2770	74-2772	*74-2773	*84-2987	84-2988
			84-2989	*84-2991	*91-3311						
DOBRK	014010	G	49-1800	49-1820	#82-2905	117-3851					
DONE	003322	G	#15-863	*45-1725	47-1745	*102-3583	*104-3605	*106-3647	*107-3682	*109-3720	*111-3751
			*113-3782	*115-3813	*117-3912						
DR	= 000060	G	#16-917	37-1628	37-1630	*37-1632	*39-1660	39-1661	*39-1663	74-2766	76-2797
			*91-3274	91-3278	91-3280	104-3600	106-3631	106-3631	106-3632	106-3632	107-3666
			107-3666	107-3667	107-3667	109-3708	109-3709	109-3709	111-3739	111-3740	111-3740
			113-3770	113-3771	113-3771	115-3801	115-38	115-3802			
DRVCHK	002220		#9-580	109-3706	111-3737	113-3768	115-37?9				
EF.CON	= 000036	G	#10-653								
EF.NEW	= 000035	G	#10-653								
EF.PWR	= 000034	G	#10-653								
EF.RES	= 000037	G	#10-653								
EF.STA	= 000040	G	#10-653	91-3242							
ENDGP8	010146		58-2089	58-2186	#58-2188						
ENDRSP	007100		50-1888	#50-1940							
ENDT9	027206		117-3916	117-3931	#117-3936						
ERRDES	013170	G	72-2700	72-2714	72-2720	72-2727	72-2738	72-2742	#74-2759	86-3051	
ERRMOD	010150		58-2153	#58-2190							
ESABO	= 177720	G	#12-747								
ESCKS	= 177757	G	#12-758	12-760	12-761	12-762	70-2615				
ESCKSM	= 177757		#12-760								
ESCMD	= 177720	G	#12-756	70-2639							
ESNCRT	= 177767	G	#12-748	70-2634							
ESNOMO	= 177737	G	#12-755	70-2610							
ESNONX	= 177770	G	#12-749	70-2644							
ESOK	= 000000	G	#12-750	70-2598							
ESPART	= 177776	G	#12-751	70-2654							
ESRD	= 177757		#12-762								
ESREC	= 177711	G	#12-757	70-2659							
ESSK	= 177740	G	#12-752	70-2628							
ESSLF	= 177777	G	#12-759								
ESTRY	= 000001	G	#12-753	70-2601							
ESWLOC	= 177765	G	#12-754	70-2649							
ESWR	= 177757		#12-761								
EVL	= 000004	G	#10-653	72-2708							
EVLTHR	002222		#9-581	72-2710							
EXOFF	007427		#56-2062	58-2177	62-2277						
EXON	007426		56-2018	#56-2061	58-2106	62-2285	62-2297				
E$END	= 002100		#5-464								
E$LOAD	= 000035		#5-464	5-507							
FLGLOC	016752	G	72-2708	*91-3320	#91-3359						
FM	015730		89-3185	89-3195	#89-3218						
FMO	015712		89-3184	#89-3215							

SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE		REFERENCES								
FTLNM	003320		#15-862	72-2731							
F$AU	= 000015		#5-464	99-3490	99-3513						
F$AUTO	= 000020		#5-464	93-3367	93-3381						
F$BGN	= 000040		#5-464	5-490	6-516	9-593	10-646	74-2759	84-2976	84-2984	88-3105
			89-3149	89-3156	89-3224	91-3234	93-3367	95-3404	97-3440	99-3490	102-3575
			102-3578	102-3580	102-3587	104-3592	104-3594	104-3614	106-3622	106-3624	106-3649
			107-3657	107-3659	107-3691	109-3697	109-3699	109-3723	111-3728	111-3730	111-3754
			113-3759	113-3761	113-3785	115-3790	115-3792	115-3816	117-3821	117-3828	117-3938
			121-3989	123-4054	123-4065	125-4108	127-4143	127-4145	127-4146	127-4146	127-4151
			127-4152								
F$CLEA	= 000007		#5-464	95-3404	95-3430						
F$DU	= 000016		#5-464	97-3440	97-3468						
F$END	= 000041		#5-464	5-464	5-464	5-464	5-464	5-464	5-464	5-464	5-464
			5-464	5-464	5-464	5-464	5-464	5-464	5-464	5-464	5-490
			9-593	10-646	74-2776	84-2980	84-2993	88-3105	89-3149	89-3208	89-3224
			91-3350	93-3381	95-3430	97-3468	99-3513	102-3575	102-3578	102-3578	102-3578
			102-3580	102-3587	102-3587	104-3592	104-3592	104-3592	104-3594	104-3614	104-3614
			106-3622	106-3622	106-3622	106-3624	106-3649	106-3649	107-3657	107-3657	107-3657
			107-3659	107-3691	107-3691	109-3697	109-3697	109-3697	109-3699	109-3723	109-3723
			111-3728	111-3728	111-3728	111-3730	111-3754	111-3754	113-3759	113-3759	113-3759
			113-3761	113-3785	113-3785	115-3790	115-3790	115-3790	115-3792	115-3816	115-3816
			117-3821	117-3821	117-3821	117-3828	117-3938	117-3938	121-3989	123-4054	123-4080
			125-4122	127-4143	127-4145	127-4146	127-4151	127-4152			
F$HARD	= 000004		#5-464	123-4065	123-4080						
F$HW	= 000013		#5-464	8-552	8-565						
F$INIT	= 000006		#5-464	91-3234	91-3350						
F$JMP	= 000050		#5-464	102-3580	104-3594	106-3624	107-3659	109-3699	111-3730	113-3761	115-3792
			117-3828								
F$MOD	= 000000		#5-464	5-490	9-593	10-646	88-3105	89-3149	89-3224	102-3575	121-3989
			123-4054	127-4143							
F$MSG	= 000011		#5-464	74-2759	74-2776						
F$PROT	= 000021		#5-464	6-516	6-520						
F$PWR	= 000017		#5-464								
F$RPT	= 000012		#5-464	89-3156	89-3208						
F$SEG	= 000003		#5-464								
F$SOFT	= 000005		#5-464	125-4108	125-4122						
F$SRV	= 000010		#5-464	84-2976	84-2980	84-2984	84-2993				
F$SUB	= 000002		#5-464								
F$SW	= 000014		#5-464	9-574	9-591						
F$TEST	= 000001		#5-464	102-3578	102-3587	104-3592	104-3614	106-3622	106-3649	107-3657	107-3691
			109-3697	109-3723	111-3728	111-3754	113-3759	113-3785	115-3790	115-3816	117-3821
			117-3938								
GBTMP	010572		*62-2261	*62-2271	*62-2281	62-2282	*62-2301	#62-2307			
GBTMP2	010574		*62-2276	62-2287	62-2289	*62-2296	62-2299	#62-2308			
GETANS	007152	G	47-1747	#54-1984							
GETHRD	016316		91-3261	#91-3265							
GETPTR	007216		#54-1996								
GETR5	017222		97-3444	#97-3469							
GTAGIN	007464		#58-2091	58-2183							
GTBYTE	010346	G	56-2026	56-2046	58-2110	58-2130	58-2161	#62-2261			
GTDOWN	010104		58-2095	58-2099	58-2114	58-2125	58-2129	58-2132	58-2136	58-2163	58-2167
			58-2169	58-2175	#58-2179						

SYMBOL VALUE REFERENCES

GTOK 007730 58-2118 #58-2138
GTPKS1 007220 G 54-1993 #56-2013
GTPKS8 007430 G 54-1989 #58-2084 58-2187
GTPTR 010344 *58-2090 58-2091 58-2180 *58-2182 #60-2235
GTUM 007672 58-2123 #58-2127
G$CNTO = 000200 #5-464
G$DELM = 000372 #5-464
G$DISP = 000003 #5-464
G$EXCP = 000400 #5-464
G$HILI = 000002 #5-464
G$LOLI = 000001 #5-464
G$NO = 000000 #5-464
G$OFFS = 000400 #5-464 123-4068 123-4069 123-4070 123-4071 123-4072 125-4109 125-4110 125-4111
125-4112 125-4113 125-4114 125-4115
G$OFSI = 000376 #5-464 123-4068 123-4069 123-4070 123-4071 123-4072 125-4109 125-4110 125-4111
125-4112 125-4113 125-4114 125-4115
G$PRMA = 000001 #5-464 123-4068 123-4069
G$PRMD = 000002 #5-464 125-4109 125-4114
G$PRML = 000000 #5-464 123-4070 123-4071 123-4072 125-4110 125-4111 125-4112 125-4113 125-4115
G$RADA = 000140 #5-464
G$RADB = 000000 #5-464
G$RADD = 000040 #5-464 125-4109 125-4114
G$RADL = 000120 #5-464 123-4070 123-4071 123-4072 125-4110 125-4111 125-4112 125-4113 125-4115
G$RADO = 000020 #5-464 123-4068 123-4069
G$XFER = 000004 #5-464
G$YES = 000010 #5-464 123-4068 123-4069 123-4070 123-4071 123-4072 125-4109 125-4110 125-4111
125-4112 125-4113 125-4114 125-4115
HARDR = 000136 G #17-954 89-3180 89-3190
HARDW = 000140 G #17-955 89-3182 89-3192
HELP = 000000 #5-449 5-459 5-481 5-499 7-522 7-537 8-559 9-584 #10-598
10-636 10-655 19-1007 19-1013 21-1029 21-1034 21-1042 21-1049 21-1054
21-1060 37-1556 37-1568 37-1573 37-1579 37-1584 37-1590 37-1598 37-1605
37-1611 37-1617 #89-3111 91-3328 91-3338 95-3411 95-3418 97-3450 97-3456
99-3493 99-3499 #100-3519 100-3560 100-3566 121-3990 121-3995 121-4005 #123-4016
123-4074 123-4090 125-4116 127-4136
HOE = 100000 G #10-653
IRD 013126 72-2732 #72-2740
HRDRD = 000016 G #11-679 68-2554
HRDWR = 000020 G #11-680 68-2556
HRD1 011746 68-2541 #68-2550
IBE = 010000 G #10-653
IDPTR 003324 G #15-864 *45-1726 45-1727 45-1729 *45-1731
IDU = 000040 G #10-653
IER = 020000 G #10-653
INIT 016164 #91-3236
INITWD 014542 *82-2905 82-2932 82-2940 82-2947 82-2954 82-2960 #84-3010
INIT2 016212 91-3243 #91-3245
ISR = 000100 G #10-653
IXE = 004000 G #10-653
I$AU = 000041 #5-464 #99-3490 #99-3513
I$AUTO = 000041 #5-464 #93-3367 #93-3381
I$CLN = 000041 #5-464 #95-3404 #95-3430

CZTUUDO CRFATED BY MACRO ON 25-FEB-82 AT 07:08 PAGE 7 SEQ 0134
SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE		REFERENCES								
I$DU	= 000041		#5-464	#97-3440	#97-3468						
I$HRD	= 000041		#123-4065	#123-4080							
I$INIT	= 000041		#5-464	#91-3234	#91-3350						
I$MOD	= 000041		#5-464	5-490	#5-490	9-593	#9-593	10-646	#10-646	88-3105	#88-3105
			89-3149	#89-3149	89-3224	#89-3224	102-3575	#102-3575	121-3989	#121-3989	123-4054
			#123-4054	127-4143	#127-4143						
I$MSG	= 000041		#5-464	#74-2759	#74-2776						
I$PROT	= 000040		#5-464	#6-516							
I$PTAB	= 000041		#5-464	127-4146	#127-4146	127-4151	#127-4151				
I$PWR	= 000041		#5-464								
I$RPT	= 000041		#5-464	#89-3156	#89-3208						
I$SEG	= 000041		#5-464	102-3578	104-3592	106-3622	107-3657	109-3697	111-3728	113-3759	115-3790
			117-3821								
I$SETU	= 000041		#5-464	127-4145	#127-4145	127-4146	127-4152	#127-4152			
I$SFT	= 000041		#125-4108	#125-4122							
I$SRV	= 000041		#5-464	#84-2976	#84-2980	#84-2984	#84-2993				
I$SUB	= 000041		#5-464	102-3578	104-3592	106-3622	107-3657	109-3697	111-3728	113-3759	115-3790
			117-3821								
I$TST	= 000041		#5-464	102-3578	#102-3578	102-3580	102-3587	#102-3587	#102-3587	104-3592	#104-3592
			104-3594	104-3614	#104-3614	#104-3614	106-3622	#106-3622	106-3624	106-3649	#106-3649
			#106-3649	107-365?	#107-3657	107-3659	107-3691	#107-3691	#107-3691	109-3697	#109-3697
			109-3699	109-3723	#109-3723	#109-3723	111-3728	#111-3728	111-3730	111-3754	#111-3754
			#111-3754	113-3759	#113-3759	113-3761	113-3785	#113-3785	#113-3785	115-3790	#115-3790
			115-3792	115-3816	#115-3816	#115-3816	117-3821	#117-3821	117-3828	117-3938	#117-3938
			#117-3938								
J$JMP	= 000167		#5-464								
LENGTH	002210		#9-576	91-3322							
LGOFST	= 000120	G	#17-944	72-2692							
LNCNT	015130		*88-3084	*88-3089	#88-3098						
LOE	= 040000	G	#10-653								
LOG	012634	G	52-:966	62-2305	66-2398	66-2418	66-2427	66-2455	68-2505	68-2528	68-2557
			70-2665	#72-2684	82-2918	82-2962	84-3006				
LOGO	013156		72-2721	72-2739	#72-2747						
LOGOK	012722		72-2699	#72-2702							
LOGOK2	013006		72-2709	72-2711	#72-2716						
LOGU	012710		#72-2700								
LOG1	013024		*72-2718	*72-2719	#72-2720	72-2720					
LOG2	013054		*72-2725	*72-2726	#72-2727	72-2727					
LOG3	013114		*72-2736	*72-2737	#72-2738	72-2738					
LOG38	013136		*72-2740	*72-2741	#72-2742	72-2742					
LOT	= 000010	G	#10-653								
LSTDEV	003366	G	#19-994	37-1634	39-1664	41-1686	45-1729	49-1807	49-1830	58-2180	64-2341
			89-3196	91-3254	93-3376	117-3930					
L$ACP	002110	G	#5-507								
L$APT	002036	G	#5-507								
L$AU	017302	G	5-507	#99-3490							
L$AUT	002070	G	#5-507								
L$AUTO	016754	G	5-507	#93-3367							
L$CCP	002106	G	#5-507								
L$CLEA	017136	G	5-507	#95-3404							
L$CO	002032	G	#5-507								
L$DEPO	002011	G	#5-507								

SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE		REFERENCES					
L$DESC	002122	G	5-507	#5-509				
L$DESP	002076	G	#5-507					
L$DEVP	002060	G	#5-507					
L$DISP	002152	G	5-507	#7-535				
L$DLY	002116	G	#5-507					
L$DTP	002040	G	#5-507					
L$DTYP	002034	G	#5-507					
L$DU	017156	G	5-507	#97-3440				
L$DUT	002072	G	#5-507					
L$DVTY	005510	G	5-507	#21-1027				
L$EF	002052	G	#5-507					
L$ENVI	002044	G	#5-507					
L$ETP	002102	G	#5-507					
L$EXP1	002046	G	#5-507					
L$EXP4	002064	G	#5-507					
L$EXP5	002066	G	#5-507					
L$HARD	041342	G	5-507	123-4065	#123-4065			
L$HIME	002120	G	#5-507					
L$HPCP	002016	G	#5-507					
L$HPTP	002022	G	#5-507					
L$HW	002176	G	5-507	8-552	#8-552			
L$ICP	002104	G	#5-507					
L$INIT	016164	G	5-507	#91-3234				
L$LADP	002026	G	#5-507					
L$LAST	042150	G	5-507	#127-4142	127-4152			
L$LOAD	002100	G	#5-507					
L$LUN	002074	G	#5-507	*72-2689	*72-2690	*86-3049	*86-3050	*91-3268
L$MREV	002050	G	#5-507					
L$NAME	002000	G	#5-507					
L$PRIO	002042	G	#5-507					
L$PROT	002142	G	5-507	#6-516				
L$PRT	002112	G	#5-507					
L$REPP	002062	G	#5-507					
L$REV	002010	G	#5-507					
L$RPT	015150	G	5-507	#89-3156				
L$SOFT	041504	G	5-507	125-4108	#125-4108			
L$SPC	002056	G	#5-507					
L$SPCP	002020	G	#5-507					
L$SPTP	002024	G	#5-507					
L$STA	002030	G	#5-507					
L$SW	002210	G	5-507	9-574	#9-574			
L$TEST	002114	G	#5-507					
L$TIML	002014	G	#5-507					
L$UNIT	002012	G	#5-507	91-3260	91-3316			
L10001	002206		8-552	#8-565				
L10002	002226		9-574	#9-591				
L10003	013360		#74-2776					
L10004	014446		#84-2980					
L10005	014502		#84-2993					
L10006	015572		#89-3208					
L10007	016662		#91-3350					
L10010	017060		#93-3381					

SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE		REFERENCES								
L10011	017154		#95-3430								
L10012	017220		#97-3468								
L10013	017302		#99-3513								
L10014	017504		102-3580	#102-3587							
L10015	017756		104-3594	#104-3614							
L10016	021350		106-3624	#106-3649							
L10017	022754		107-3659	#107-3691							
L10020	023744		109-3699	#109-3723							
L10021	024530		111-3730	#111-3754							
L10022	025520		113-3761	#113-3785							
L10023	026304		115-3792	#115-3816							
L10024	027210		117-3828	#117-3938							
L10025	041404		123-4065	#123-4080							
L10026	041566		125-4108	#125-4122							
L10027	042154		#127-4146								
L10031	042164		127-4146	#127-4151							
MABEE	013066		72-2724	#72-2730							
MESMRS	010200		58-2191	#58-2194							
MODRSP	010252		58-2155	#58-2196							
MRSDLY	010254		*58-2144	*58-2149	#58-2198						
MRSP	= 000210	G	#17-976	50-1887	58-2087	58-2174	64-2338	*66-2409	*66-2449	*91-3312	117-3923
MSAGE1	027212		117-3918	#117-3940							
MSAGE2	027253		117-3925	#117-3941							
MSAGE3	027307		117-3927	#117-3942							
MSAGE4	027336		117-3929	#117-3943							
MSAUTO	017116		93-3389	#93-3394							
MSBDA	002340	G	#14-801	86-3051							
MSCMD	002704	G	13-786	#14-823							
MSCOM	002404	G	13-776	#14-805							
MSG1	041404		123-4068	#123-4082							
MSG1B	041415		123-4069	#123-4083							
MSG1C	041432		123-4070	#123-4084							
MSG2	041450		123-4071	#123-4085							
MSG3	041465		123-4072	#123-4086							
MSG4	041566		125-4109	#125-4123							
MSG4B	041633		125-4110	#125-4124							
MSG5	041675		125-4111	#125-4125							
MSG6	041727		125-4112	#125-4126							
MSG7	041754		125-4113	#125-4127							
MSG8	042002		125-4114	#125-4128							
MSG9	042043		125-4115	#125-4129							
MSHCHK	002556	G	13-779	#14-815							
MSHDRD	003154	G	13-777	#14-839							
MSHDWR	003216	G	13-778	#14-841							
MSNIT	002620	G	13-783	#14-817							
MSNLOG	002322	G	13-770	13-791	#14-799						
MSNOMO	002446	G	13-782	#14-807							
MSNOTP	002464	G	13-792	#14-809							
MSNRSP	002764	G	13-790	#14-829							
MSOVRN	003260	G	13-775	#14-843							
MSPART	002634	G	13-784	#14-819							
MSQRSP	003000	G	13-774	#14-831							

SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE		REFERENCES								
MSREC	002720	G	13-787	#14-825							
MSRNIT	002536	G	13-773	#14-813							
MSSELF	002364	G	13-788	#14-803							
MSSFRD	003054	G	13-771	#14-835							
MSSFWR	003114	G	13-772	#14-837							
MSSKER	002306	G	13-780	#14-797							
MSTOSN	003032	G	13-793	#14-833							
MSUNIT	002656	G	13-785	#14-821							
MSWPRO	002514	G	13-781	#14-811							
MSWRSP	002740	G	13-789	#14-827							
MXRTRY	003330	G	#15-866	68-2540							
NCARS	= 000054	G	#11-693	70-2636							
NODRVS	016714		91-3282	#91-3355							
NOMOR	006534		49-1836	#49-1843							
NOMOT	= 000030	G	#11-684	70-2612							
NOREE	011334		68-2491	#68-2495							
NOUNIT	= 000036	G	#11-687	70-2646							
NOXOFF	006632		#50-1889								
NTSFT	013036		72-2717	#72-2723							
NXTRET	006530		49-1806	49-1835	#49-1840						
NXTST	006102	G	47-1743	#49-1775							
NXTST2	006352		49-1808	#49-1812							
ODTFLG	= ******		91-3237	127-4153							
ONEFIL	= 000001		#2-4	2-8	4-445	5-446	5-485	9-594	10-595	10-608	88-3107
			89-3108	89-3121	99-3515	100-3516	100-3527	122-4011	123-4012	123-4026	
OTL	= 000010	G	#11-676	66-2414							
OVRFLO	013534		72-2700	#74-2781							
OVRN	= 000012	G	#11-677	66-2417							
O$APTS	= 000000		#5-464	5-507							
O$AU	= 000001		#5-464	#5-497	5-507						
O$BGNR	= 000001		#5-464	#5-497	5-507						
O$BGNS	= 000001		#5-464	#5-497	5-507						
O$DU	= 000001		#5-464	#5-497	5-507						
O$ERRT	= 000000		#5-464	5-507							
O$GNSW	= 000001		#5-464	#5-497	5-507						
O$POIN	= 000001		#5-464	#5-497	#5-497	#5-497	#5-497	#5-497	#5-497	5-497	5-507
O$SETU	= 000001		#5-464	#5-497	5-507	127-4142					
PARTL	= 000034	G	#11-686	70-2656							
PATTEN	= 000072	G	#16-923	74-2768	86-3042	*91-3307	*106-3630	106-3631	*107-3665	107-3666	*109-3705
			*109-3708	109-3709	*111-3736	*111-3739	*113-3767	*113-3770	113-3771	*115-3798	*115-3801
PDTFLG	016750	G	*91-3276	91-3303	#91-3358						
PERDEV	006364		#49-1814	49-1833							
PKPTR	= 000104	G	#16-929	*50-1903	58-2109	58-2168	*58-2170	*66-2384	86-3034	88-3083	
PNT	= 001000	G	#10-653								
PPSOT9	002224		#9-582	117-3915							
PRBUF	002214		#9-578	88-3081							
PRDAT	015132		*88-3085	88-3086	#88-3099						
PRFORM	015134		88-3086	#88-3100							
PRI	= 002000	G	#10-653								
PRI00	= 000000	G	#10-653	82-2926							
PRI01	= 000040	G	#10-653								
PRI02	= 000100	G	#10-653								

SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE		REFERENCES								
PRI03	= 000140	G	#10-653								
PRI04	= 000200	G	#10-653								
PRI05	= 000240	G	#10-653								
PRI06	= 000300	G	#10-653								
PRI07	= 000340	G	5-507	#10-653	82-2927	82-2929	93-3369				
PRNPAK	014764	G	86-3055	#88-3076							
PRNSIZ	003336	G	#15-869	*86-3054	*88-3087						
PTR	017252		*97-3469	97-3470	*97-3473	#97-3476					
RCBCNT	003316		#15-861	*56-2024	*56-2035	*56-2040	*56-2044	*58-2103	*58-2122	*58-2126	*58-2128
			*58-2159	*66-2386	66-2393	66-2460					
RCBFSZ	= 001036	G	#12-731	43-1704	121-3974	121-3978	121-3979	121-3980	121-3981	121-3982	121-3983
			121-3984	121-3985							
RCDB	= 000024	G	#16-909	62-2265	62-2290	82-2925	84-2987	*91-3299			
RCFLG	003314	G	#15-860	*56-2025	56-2031	*58-2101	58-2117	*66-2387			
RCINIT	= 000006	G	#11-675	66-2426							
RCSR	= 000022	G	#16-908	58-2152	62-2263	62-2279	82-2949	82-2966	84-2986	*91-3297	93-3374
RCVBUF	= 000102	G	#16-928	43-1703	50-1903	56-2021	66-2378	*91-3252			
RCVHND	014450		#84-2986								
RCVINT	014450	G	82-2927	#84-2984							
RDN0	= 000114	G	#16-933	*50-1921	89-3185						
RDN1	= 000116	G	#16-934	*50-1923	89-3195						
REC	= 000064	G	#16-919	50-1899	74-2767	*91-3309	*104-3598	104-3600	*106-3629	106-3631	106-3631
			106-3632	106-3632	*106-3638	106-3639	*106-3642	*106-3643	*107-3664	107-3666	107-3666
			107-3667	107-3667	*107-3673	107-3674	*107-3677	*107-3678	*109-3702	109-3705	109-3709
			109-3709	*109-3712	*109-3717	*111-3733	111-3736	111-3740	111-3740	*111-3743	*111-3748
			*113-3764	113-3767	113-3771	113-3771	*113-3774	*113-3779	*115-3795	115-3798	115-3802
			115-3802	*115-3805	*115-3810	117-3862	117-3889				
RECDAT	017742		104-3598	104-3603	#104-3608						
RECERR	= 000042	G	#11-689	70-2661							
RECID	013442	G	74-2769	#74-2779							
RECID2	013616		74-2772	#74-2783							
RECOV	012044		68-2517	#68-2568							
RETERR	012246		68-2536	#68-2578							
RETRY	= 000002	G	#16-900	*68-2511	68-2512	68-2517	*68-2529	*68-2537	68-2540	*68-2542	68-2543
			*68-2558	*91-3308							
RLUN	015574		*89-3174	*89-3175	89-3184	#89-3209					
RPTR	015576		*89-3165	89-3170	89-3196	*89-3198	#89-3210				
RSCMND	= 000002	G	#12-711	12-717	50-1909	102-3582	104-3600	104-3600	106-3631	106-3631	106-3632
			106-3632	107-3666	107-3666	107-3667	107-3667	109-3709	109-3709	111-3740	111-3740
			113-3771	113-3771	115-3802	115-3802	117-3832	117-3859	117-3875	117-3886	117-3902
RSCONT	= 000020	G	#12-712	58-2196	66-2390	82-2954	106-3631	106-3631	107-3666	107-3666	109-3709
			109-3709	113-3771	113-3771						
RSDASZ	= 000204	G	#12-724	12-726	12-731	56-2040	66-2431	66-2438			
RSDATA	= 000001	G	#12-716	56-2038	58-2124	58-2168	66-2405	66-2429	106-3631	106-3631	106-3632
			106-3632	107-3666	107-3666	107-3667	107-3667	109-3709	109-3709	111-3740	111-3740
			113-3771	113-3771	115-3802	115-3802	117-3845				
RSDNSZ	= 000222	G	#12-726	58-2126							
RSEND	= 000002	G	#12-717	56-2033	58-2120	66-2400	66-2441	102-3582	106-3631	106-3631	106-3632
			106-3632	107-3666	107-3666	107-3667	107-3667	109-3709	109-3709	111-3740	111-3740
			113-3771	113-3771	115-3802	115-3802					
RSGCDP	= 000034	G	#12-728	117-3846							
RSINIT	= 000004	G	#12-715	66-2423	84-3010						

SYMBOL CROSS REFERENCE
CREF

SYMBOL	VALUE		REFERENCES								
RSMSIZ	= 000012	G	#12-722	12-730	102-3582	102-3582	104-3600	104-3600	106-3631	106-3631	106-3631
			106-3631	106-3632	106-3632	106-3632	106-3632	107-3666	107-3666	107-3666	107-3666
			107-3667	107-3667	107-3667	107-3667	109-3709	109-3709	109-3709	109-3709	111-3740
			111-3740	111-3740	111-3740	113-3771	113-3771	113-3771	113-3771	115-3802	115-3802
			115-3802	115-3802	117-3833	117-3840	117-3860	117-3868	117-3887	117-3895	
RSNDSZ	= 000016	G	#12-720	12-726	12-731	56-2035	58-2122	102-3582	104-3600	106-3631	106-3631
			106-3632	106-3632	107-3666	107-3666	107-3667	107-3667	109-3709	109-3709	111-3740
			111-3740	113-3771	113-3771	115-3802	115-3802	117-3876	117-3903		
RSNTAB	002226		#13-770	72-2705							
RSSEND	= 000100	G	#12-736								
RSSGET	= 000012	G	#12-740	117-3834							
RSSNIT	= 000001	G	#12-742	82-2908	117-3861						
RSSNOP	= 000000	G	#12-741	117-3888							
RSSNSZ	= 000016	G	#12-730	66-2443	102-3582	104-3600	106-3631	106-3631	106-3632	106-3632	107-3666
			107-3666	107-3667	107-3667	109-3709	109-3709	111-3740	111-3740	113-3771	113-3771
			115-3802	115-3802	117-3842	117-3874	117-3901				
RSSRD	= 000002	G	#12-738	50-1916	70-2603	106-3631	106-3632	106-3632	107-3666	107-3667	107-3667
			109-3709	111-3740	111-3740	113-3771	115-3802	115-3802			
RSSSEK	= 000005	G	#12-739	104-3600							
RSSSLF	= 000007	G	#12-743	70-2620	102-3582						
RSSWR	= 000003	G	#12-737	50-1925	70-2607	106-3631	107-3666	109-3709	113-3771		
RSVP	006560	G	#50-1880	102-3582	104-3600	106-3631	106-3631	106-3631	106-3632	106-3632	107-3666
			107-3666	107-3666	107-3667	107-3667	109-3709	109-3709	109-3709	111-3740	111-3740
			113-3771	113-3771	113-3771	115-3802	115-3802	117-3849	117-3879	117-3906	
RSXOFF	= 000023	G	#12-714	56-2062	121-3971	121-3972					
RSXON	= 000020	G	#12-713	56-2061							
RTRYN	012204		68-2512	68-2543	#68-2574						
RUN	006052	G	#47-1743	47-1752	102-3579	102-3579	104-3593	104-3593	106-3623	106-3623	107-3658
			107-3658	109-3698	109-3698	111-3729	111-3729	113-3760	113-3760	115-3791	115-3791
			117-3825								
SAVCNT	= 000206	G	#17-975	49-1787	*49-1798						
SECREC	003334	G	#15-868	*91-3324	*91-3327	109-3717	111-3748	113-3779	115-3810		
SERVST	010342		*54-1987	58-2184	*60-2220	#60-2234					
SETDR	005632	G	#39-1658	102-3579	104-3593	106-3623	107-3658	109-3698	111-3729	113-3760	115-3791
SETLEN	016624		#91-3322								
SETPTR	005706		*39-1658	39-1659	39-1664	*39-1666	#39-1669				
SETSRV	010256		58-2094	58-2098	#60-2210						
SETUP	006004	G	#45-1725	102-3579	102-3579	104-3593	104-3593	106-3623	106-3623	107-3658	107-3658
			109-3698	109-3698	111-3729	111-3729	113-3760	113-3760	115-3791	115-3791	117-3824
SFPTBL	002210	G	#9-574								
SFT	013104		72-2733	#72-2736							
SFTOUT	041566		#125-4122								
SFTRD	= 000002	G	#11-673	68-2525							
SFTWR	= 000004	G	#11-674	68-2527							
SKERR	= 000024	G	#11-682	70-2630							
SLFER	= 000044	G	#11-690	70-2625							
SND	006650		50-1890	#50-1893	50-1897						
SNDBYT	007102	G	50-1893	#52-1955	56-2019	58-2108	58-2156	58-2178	62-2278	62-2286	62-2298
SNDCNT	= 000070	G	#16-922	*50-1892	*50-1896	*102-3582	*104-3600	*106-3631	*106-3631	*106-3631	*106-3631
			*106-3632	*106-3632	*107-3666	*107-3666	*107-3666	*107-3666	*107-3667	*107-3667	*109-3709
			*109-3709	*109-3709	*109-3709	*111-3740	*111-3740	*113-3771	*113-3771	*113-3771	*113-3771
			*115-3802	*115-3802	*117-3842	*117-3874	*117-3901				

```
SYMBOL  VALUE              REFERENCES
SNDHND    014434           #84-2978
SNDINT    014434   G        82-2929   #84-2976
SOFTR   = 000122   G       #17-945     89-3176    89-3186
SOFTW   = 000124   G       #17-946     89-3178    89-3188
SRVTBL    010322            60-2214   #60-2225
STAEOP    002212           #9-577      95-3407
STATHD    015600            89-3166   #89-3212
STATUS  = 000000   G       #16-899     117-3921
STHD2     016054            89-3168   #89-3221
STRT      016746   G       *91-3240   *91-3244    91-3285   #91-3357
SUCCS   = 000076   G       #16-925    *50-1901    68-2499    68-2515   *70-2597    74-2769   *91-3310
SUCOTL  = 000046   G       #11-691     70-2664
SVCGBL  = 000000           #5-464     #5-473      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-509      6-516      7-535      8-552      8-552
                            9-574      9-574      21-1027    74-2759    84-2976    84-2984    89-3156    91-3234    93-3367
                            95-3404    97-3440    99-3490    123-4065   125-4108  #127-4142   127-4142   127-4142   127-4142
SVCINS  = 000001           #5-464     #5-470      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                            5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-509
                            5-509      5-509      5-509      5-509      5-509      7-535      7-535      7-535      7-535
                            7-535      7-535      7-535      7-535      7-535      7-535      7-535      7-535      7-535
                            7-535      7-535      7-535      7-535      7-535      7-535      7-535      7-535      7-535
                            7-535      7-535      7-535      7-535      7-535      7-535      7-535      7-535      8-552
                            8-552      8-552      9-574      9-574      9-574      21-1027    21-1027    21-1027    21-1027
                            21-1027    21-1027    47-1749    47-1749    47-1749    49-1836    49-1836    49-1836    49-1836
                            49-1836    49-1836    49-1836    49-1836    49-1836    49-1836    49-1836    49-1836    49-1837
                            49-1837    49-1837    49-1839    49-1839    49-1839    52-1960    52-1960    52-1960    58-2191
                            58-2191    58-2191    58-2191    58-2191    58-2191    58-2191    58-2191    58-2191    58-2191
                            58-2191    58-2191    58-2191    58-2191    58-2191    58-2191    58-2191    58-2191    62-2284
                            62-2284    62-2284    68-2512    68-2512    68-2512    68-2512    68-2512    68-2512    68-2512
                            68-2512    68-2512    68-2512    68-2512    68-2512    68-2512    68-2512    68-2512    68-2512
                            68-2512    68-2512    68-2517    68-2517    68-2517    68-2517    68-2517    68-2517    68-2517
                            68-2517    68-2517    68-2517    68-2517    68-2517    68-2517    68-2517    68-2517    68-2517
                            68-2517    68-2517    68-2520    68-2520    68-2520    68-2520    68-2520    68-2520    68-2520
                            68-2520    68-2520    68-2520    68-2520    68-2520    68-2520    68-2520    68-2520    68-2522
                            68-2522    68-2522    68-2522    68-2522    68-2522    68-2522    68-2522    68-2522    68-2522
```

SYMBOL VALUE REFERENCES

68-2522 68-2522 68-2522 68-2522 68-2522 68-2536 68-2536 68-2536 68-2536
68-2536 68-2536 68-2536 68-2536 68-2536 68-2536 68-2536 68-2536 68-2536
68-2536 68-2536 68-2543 68-2543 68-2543 68-2543 68-2543 68-2543 68-2543
68-2543 68-2543 68-2543 68-2543 68-2543 68-2543 68-2543 68-2543 68-2543
68-2543 68-2543 68-2551 68-2551 68-2551 68-2551 68-2551 68-2551 68-2551
68-2551 68-2551 68-2551 68-2551 68-2551 68-2551 68-2551 68-2551 72-2700
72-2700 72-2700 72-2700 72-2700 72-2700 72-2700 72-2700 72-2700 72-2700
72-2700 72-2700 72-2714 72-2714 72-2714 72-2714 72-2714 72-2714 72-2714
72-2714 72-2714 72-2714 72-2714 72-2714 72-2720 72-2720 72-2720 72-2720
72-2720 72-2720 72-2720 72-2720 72-2720 72-2720 72-2720 72-2720 72-2727
72-2727 72-2727 72-2727 72-2727 72-2727 72-2727 72-2727 72-2727 72-2727
72-2727 72-2727 72-2738 72-2738 72-2738 72-2738 72-2738 72-2738 72-2738
72-2738 72-2738 72-2738 72-2738 72-2738 72-2738 72-2742 72-2742 72-2742
72-2742 72-2742 72-2742 72-2742 72-2742 72-2742 72-2742 72-2742 72-2742
72-2746 72-2746 74-2766 74-2766 74-2766 74-2766 74-2766 74-2766 74-2766
74-2766 74-2766 74-2766 74-2766 74-2766 74-2766 74-2766 74-2766 74-2766
74-2766 74-2766 74-2766 74-2766 74-2766 74-2766 74-2766 74-2766 74-2766
74-2766 74-2766 74-2766 74-2766 74-2766 74-2769 74-2769 74-2769 74-2769
74-2769 74-2769 74-2769 74-2769 74-2769 74-2769 74-2769 74-2769 74-2769
74-2769 74-2769 74-2769 74-2769 74-2769 74-2769 74-2769 74-2769 74-2769
74-2769 74-2769 74-2769 74-2769 74-2769 74-2769 74-2769 74-2769 74-2769
74-2772 74-2772 74-2772 74-2772 74-2772 74-2772 74-2772 74-2772 74-2772
74-2772 74-2772 74-2772 74-2772 74-2772 74-2776 74-2776 74-2776 82-2911
82-2911 82-2911 82-2926 82-2926 82-2926 82-2926 82-2926 82-2926 82-2927
82-2927 82-2927 82-2927 82-2927 82-2927 82-2927 82-2927 82-2927 82-2927
82-2927 82-2927 82-2927 82-2927 82-2927 82-2927 82-2927 82-2927 82-2929
82-2929 82-2929 82-2929 82-2929 82-2929 82-2929 82-2929 82-2929 82-2929
82-2929 82-2929 82-2929 82-2929 82-2929 82-2929 82-2929 82-2929 82-2967
82-2967 82-2967 82-2967 82-2967 82-2967 82-2969 82-2969 82-2969 82-2969
82-2969 82-2969 84-2980 84-2980 84-2980 84-2993 84-2993 84-2993 84-3001
84-3001 84-3001 84-3002 84-3002 84-3002 86-3051 86-3051 86-3051 86-3051
86-3051 86-3051 86-3051 86-3051 86-3051 86-3051 86-3051 86-3051 86-3052
86-3052 86-3052 86-3052 86-3052 86-3052 86-3052 86-3052 86-3052 86-3052
86-3052 86-3052 86-3052 86-3052 86-3052 86-3052 86-3052 86-3052 88-3086
88-3086 88-3086 88-3086 88-3086 88-3086 88-3086 88-3086 88-3086 88-3086
88-3086 88-3086 88-3086 88-3086 88-3086 88-3086 88-3086 88-3086 88-3086
88-3086 88-3086 88-3091 88-3091 88-3091 88-3091 88-3091 88-3091 88-3091
88-3091 88-3091 88-3091 88-3091 88-3091 88-3091 88-3091 88-3091 88-3093
88-3093 88-3093 88-3093 88-3093 88-3093 88-3093 88-3093 88-3093 88-3093
88-3093 88-3093 88-3093 88-3093 88-3093 89-3164 89-3164 89-3164 89-3166
89-3166 89-3166 89-3166 89-3166 89-3166 89-3166 89-3166 89-3166 89-3166
89-3166 89-3166 89-3166 89-3166 89-3166 89-3167 89-3167 89-3167 89-3168
89-3168 89-3168 89-3168 89-3168 89-3168 89-3168 89-3168 89-3168 89-3168
89-3168 89-3168 89-3168 89-3168 89-3168 89-3169 89-3169 89-3169 89-3184
89-3184 89-3184 89-3184 89-3184 89-3184 89-3184 89-3184 89-3184 89-3184
89-3184 89-3184 89-3184 89-3184 89-3184 89-3184 89-3184 89-3184 89-3185
89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185
89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185
89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185
89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185 89-3185
89-3185 89-3185 89-3185 89-3185 89-3185 89-3195 89-3195 89-3195 89-3195

SYMBOL VALUE REFERENCES

89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195
89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195
89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195
89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195 89-3195
89-3195 89-3195 89-3208 89-3208 89-3208 91-3242 91-3242 91-3242 91-3242
91-3242 91-3242 91-3243 91-3243 91-3243 91-3262 91-3262 91-3262 91-3262
91-3262 91-3262 91-3262 91-3262 91-3262 91-3262 91-3262 91-3262 91-3263
91-3263 91-3263 91-3269 91-3269 91-3269 91-3269 91-3269 91-3269 91-3269
91-3269 91-3269 91-3270 91-3270 91-3270 91-3282 91-3282 91-3282 91-3282
91-3282 91-3282 91-3282 91-3282 91-3282 91-3282 91-3282 91-3282 91-3283
91-3283 91-3283 91-3320 91-3320 91-3320 91-3320 91-3320 91-3320 91-3350
91-3350 91-3350 93-3369 93-3369 93-3369 93-3369 93-3369 93-3369 93-3369
93-3369 93-3369 93-3369 93-3369 93-3369 93-3369 93-3369 93-3369 93-3369
93-3369 93-3369 93-3380 93-3380 93-3380 93-3380 93-3380 93-3380 93-3381
93-3381 93-3381 93-3389 93-3389 93-3389 93-3389 93-3389 93-3389 93-3389
93-3389 93-3389 93-3389 93-3389 93-3389 93-3389 93-3389 93-3389 93-3392
93-3392 93-3392 95-3409 95-3409 95-3409 95-3430 95-3430 95-3430 97-3448
97-3448 97-3448 97-3448 97-3448 97-3448 97-3448 97-3448 97-3448 97-3448
97-3448 97-3448 97-3448 97-3448 97-3448 97-3448 97-3448 97-3448 97-3468
97-3468 97-3468 99-3513 99-3513 99-3513 102-3580 102-3580 102-3580 102-3580
102-3580 102-3580 102-3587 102-3587 102-3587 104-3594 104-3594 104-3594 104-3594
104-3594 104-3594 104-3614 104-3614 104-3614 106-3624 106-3624 106-3624 106-3624
106-3624 106-3624 106-3649 106-3649 106-3649 107-3659 107-3659 107-3659 107-3659
107-3659 107-3659 107-3691 107-3691 107-3691 109-3699 109-3699 109-3699 109-3699
109-3699 109-3699 109-3723 109-3723 109-3723 111-3730 111-3730 111-3730 111-3730
111-3730 111-3730 111-3754 111-3754 111-3754 113-3761 113-3761 113-3761 113-3761
113-3761 113-3761 113-3785 113-3785 113-3785 115-3792 115-3792 115-3792 115-3792
115-3792 115-3792 115-3816 115-3816 115-3816 117-3828 117-3828 117-3828 117-3828
117-3828 117-3828 117-3918 117-3918 117-3918 117-3918 117-3918 117-3918 117-3918
117-3918 117-3918 117-3918 117-3918 117-3918 117-3918 117-3918 117-3918 117-3925
117-3925 117-3925 117-3925 117-3925 117-3925 117-3925 117-3925 117-3925 117-3925
117-3925 117-3925 117-3925 117-3925 117-3925 117-3925 117-3925 117-3925 117-3927
117-3927 117-3927 117-3927 117-3927 117-3927 117-3927 117-3927 117-3927 117-3927
117-3927 117-3927 117-3927 117-3927 117-3927 117-3927 117-3927 117-3927 117-3929
117-3929 117-3929 117-3929 117-3929 117-3929 117-3929 117-3929 117-3929 117-3929
117-3929 117-3929 117-3929 117-3929 117-3929 117-3929 117-3929 117-3929 117-3938
117-3938 117-3938 123-4065 123-4065 123-4065 123-4068 123-4068 123-4068 123-4068
123-4068 123-4068 123-4068 123-4068 123-4068 123-4068 123-4068 123-4068 123-4069
123-4069 123-4069 123-4069 123-4069 123-4069 123-4069 123-4069 123-4069 123-4069
123-4069 123-4069 123-4070 123-4070 123-4070 123-4070 123-4070 123-4070 123-4070
123-4070 123-4070 123-4071 123-4071 123-4071 123-4071 123-4071 123-4071 123-4071
123-4071 123-4071 123-4072 123-4072 123-4072 123-4072 123-4072 123-4072 123-4072
123-4072 123-4072 123-4080 123-4080 123-4080 125-4108 125-4108 125-4108 125-4109
125-4109 125-4109 125-4109 125-4109 125-4109 125-4109 125-4109 125-4109 125-4109
125-4109 125-4109 125-4109 125-4109 125-4109 125-4110 125-4110 125-4110 125-4110
125-4110 125-4110 125-4110 125-4110 125-4110 125-4111 125-4111 125-4111 125-4111
125-4111 125-4111 125-4111 125-4111 125-4111 125-4112 125-4112 125-4112 125-4112
125-4112 125-4112 125-4112 125-4112 125-4112 125-4113 125-4113 125-4113 125-4113
125-4113 125-4113 125-4113 125-4113 125-4113 125-4114 125-4114 125-4114 125-4114
125-4114 125-4114 125-4114 125-4114 125-4114 125-4114 125-4114 125-4114 125-4114
125-4114 125-4114 125-4115 125-4115 125-4115 125-4115 125-4115 125-4115 125-4115
125-4115 125-4115 125-4122 125-4122 125-4122 127-4142 127-4142 127-4142 127-4142

SYMBOL CROSS REFERENCE
CREF

SYMBOL	VALUE		REFERENCES								
			127-4142	127-4142	127-4142	127-4142	127-4142	127-4146	127-4146	127-4146	127-4146
			127-4146	127-4146							
SVCSUB	= 000001		#5-464	#5-472							
SVCTAG	= 000001		#5-464	#5-474	8-565	9-591	74-2776	84-2980	84-2993	89-3208	91-3350
			93-3381	95-3430	97-3468	99-3513	102-3587	104-3614	106-3649	107-3691	109-3723
			111-3754	113-3785	115-3816	117-3938	123-4080	125-4122	127-4146	127-4151	
SVCTST	= 000001		#5-464	#5-471	102-3578	104-3592	106-3622	107-3657	109-3697	111-3728	113-3759
			115-3790	117-3821							
SWAPDR	005530	G	#37-1623	102-3579	104-3593	106-3623	107-3658	109-3698	111-3729	113-3760	115-3791
SWPTR	005630		*37-1624	37-1625	37-1634	*37-1636	#37-1644				
SYSTAT	003306	G	#15-849	*49-1789	*49-1794	*49-1803	54-1985	*56-2030	*58-2116	64-2326	*66-2385
			68-2490	68-2492	*68-2561	74-2766	*78-2818	*78-2821	78-2832	*82-2960	*91-3319
S$LSYM	= 010000		#5-464	#8-565	#9-591	#74-2776	#84-2980	#84-2993	#89-3208	#91-3350	#93-3381
			#95-3430	#97-3468	#99-3513	#102-3587	#104-3614	#106-3649	#107-3691	#109-3723	#111-3754
			#113-3785	#115-3816	#117-3938	#123-4080	#125-4122				
TAPLEN	003310	G	#15-858	*91-3322	*91-3323	91-3325	109-3703	109-3718	111-3734	111-3749	113-3765
			113-3780	115-3796	115-3811						
TEST9	003342	G	#15-871	50-1885	58-2085	58-2104	58-2141	58-2172	64-2336	66-2407	66-2447
			*91-3241	*117-3830	*117-3856	*117-3913					
THRSHI	012152		68-2522	#68-2572							
THRSLO	012124		68-2520	#68-2570							
TMP	= 000066	G	#16-921	*106-3627	106-3643	*106-3644	*107-3662	107-3678	*107-3679	*109-3703	*109-3710
			*109-3718	*111-3734	*111-3741	*111-3749	*113-3765	*113-3772	*113-3780	*115-3796	*115-3803
			*115-3811								
TOMANY	016664		91-3262	#91-3353							
TORCVB	= 000050	G	#11-692	62-2304	84-3005						
TOSNDB	= 000056	G	#11-694	52-1965	82-2917						
TRBUF	027722		50-1889	50-1891	50-1898	102-3582	104-3600	106-3631	106-3631	106-3631	106-3631
			106-3632	106-3632	107-3666	107-3666	107-3666	107-3666	107-3667	107-3667	109-3709
			109-3709	109-3709	109-3709	111-3740	111-3740	113-3771	113-3771	113-3771	113-3771
			115-3802	115-3802	117-3831	117-3858	117-3885	#121-3974			
TRK	= 000062	G	#16-918	*109-3704	109-3714	*109-3716	*111-3735	111-3745	*111-3747	*113-3766	113-3776
			*113-3778	*115-3797	115-3807	*115-3809					
TRPHND	017064		93-3369	#93-3389							
TRPPTR	017062		*93-3370	93-3371	93-3376	*93-3378	#93-3382				
TSTPC	= 000020	G	#16-907	*45-1728	49-1805	49-1826	*50-1881				
TSTTOP	003326		#15-865	45-1728	*102-3579	*104-3593	*106-3623	*107-3658	*109-3698	*111-3729	*113-3760
			*115-3791	*117-3823							
TST1	017350		102-3579	#102-3582							
TST2	017552		104-3593	#104-3597							
TST3	020024		106-3623	#106-3627							
TST3PT	022742		106-3630	106-3634	107-3665	107-3669	#107-3684				
TST4	021416		107-3658	#107-3662							
TST5	023022		109-3698	#109-3702							
TST5EX	023736		109-3715	#109-3720							
TST6	024012		111-3729	#111-3733							
TST6EX	024522		111-3746	#111-3751							
TST7	024576		113-3760	#113-3764							
TST7EX	025512		113-3777	#113-3782							
TST8	025566		115-3791	#115-3795							
TST8EX	026276		115-3808	#115-3813							
TST9	026330		117-3823	#117-3830							

SYMBOL	VALUE	REFERENCES								
TUVECT	= 000204 G	#17-974	82-2927	*82-2928	82-2929	*82-2930	82-2967	*82-2968	82-2969	*82-2970
		*91-3273								
T$ARGC	= 000002	#5-507	5-507	#5-507	5-507	5-507	#5-507	5-507	5-507	#5-507
		5-507	5-507	#5-507	5-507	5-507	#5-507	5-507	5-507	#58-2191
		58-2191	#58-2191	58-2191	58-2191	#68-2512	68-2512	#68-2512	68-2512	68-2512
		#68-2517	68-2517	#68-2517	68-2517	68-2517	#68-2520	68-2520	68-2520	#68-2522
		68-2522	68-2522	#68-2536	68-2536	68-2536	#68-2543	68-2543	#68-2543	68-2543
		68-2543	#68-2551	68-2551	68-2551	#74-2766	74-2766	#74-2766	74-2766	#74-2766
		74-2766	#74-2766	74-2766	74-2766	#74-2769	74-2769	#74-2769	74-2769	#74-2769
		74-2769	#74-2769	74-2769	#74-2769	74-2769	74-2769	#74-2772	74-2772	#74-2772
		74-2772	74-2772	#86-3052	86-3052	#86-3052	86-3052	86-3052	#88-3086	88-3086
		#88-3086	88-3086	88-3086	#88-3091	88-3091	88-3091	#88-3093	88-3093	88-3093
		#89-3166	89-3166	89-3166	#89-3168	89-3168	89-3168	#89-3184	89-3184	#89-3184
		89-3184	89-3184	#89-3185	89-3185	#89-3185	89-3185	#89-3185	89-3185	#89-3185
		89-3185	#89-3185	89-3185	#89-3185	89-3185	#89-3185	89-3185	#89-3185	89-3185
		#89-3185	89-3185	89-3185	#89-3195	89-3195	#89-3195	89-3195	#89-3195	89-3195
		#89-3195	89-3195	#89-3195	89-3195	#89-3195	89-3195	#89-3195	89-3195	#89-3195
		89-3195	#89-3195	89-3195	89-3195	#93-3389	93-3389	93-3389	#97-3448	97-3448
		#97-3448	97-3448	97-3448	#117-3918	117-3918	117-3918	#117-3925	117-3925	#117-3925
		117-3925	117-3925	#117-3927	117-3927	#117-3927	117-3927	117-3927	#117-3929	117-3929
		#117-3929	117-3929	117-3929						
T$CODE	= 006130	#123-4068	123-4068	#123-4068	123-4068	#123-4068	123-4068	#123-4069	123-4069	#123-4069
		123-4069	#123-4069	123-4069	#123-4070	123-4070	#123-4070	123-4070	#123-4070	123-4070
		#123-4071	123-4071	#123-4071	123-4071	#123-4071	123-4071	#123-4072	123-4072	#123-4072
		123-4072	#123-4072	123-4072	#125-4109	125-4109	#125-4109	125-4109	#125-4109	125-4109
		#125-4110	125-4110	#125-4110	125-4110	#125-4110	125-4110	#125-4111	125-4111	#125-4111
		125-4111	#125-4111	125-4111	#125-4112	125-4112	#125-4112	125-4112	#125-4112	125-4112
		#125-4113	125-4113	#125-4113	125-4113	#125-4113	125-4113	#125-4114	125-4114	#125-4114
		125-4114	#125-4114	125-4114	#125-4115	125-4115	#125-4115	125-4115	#125-4115	125-4115
T$ERRN	= 000146	#5-464	#49-1836	49-1836	#72-2700	72-2700	#72-2714	72-2714	#72-2720	72-2720
		#72-2727	72-2727	#72-2738	72-2738	#72-2742	72-2742	#86-3051	86-3051	#91-3262
		91-3262	#91-3282	91-3282						
T$EXCP	= 000000	#123-4068	123-4068	#123-4069	123-4069	#125-4109	125-4109	#125-4114	125-4114	
T$FLAG	= 000040	#102-3580	#102-3580	102-3580	102-3580	#104-3594	#104-3594	104-3594	104-3594	#106-3624
		#106-3624	106-3624	106-3624	#107-3659	#107-3659	107-3659	107-3659	#109-3699	#109-3699
		109-3699	109-3699	#111-3730	#111-3730	111-3730	111-3730	#113-3761	#113-3761	113-3761
		113-3761	#115-3792	#115-3792	115-3792	115-3792	#117-3828	#117-3828	117-3828	117-3828
T$FREE	= 042164	127-4142	#127-4152							
T$GMAN	= 000000	#5-464								
T$HILI	= 000376	#123-4068	123-4068	#123-4069	123-4069	#125-4109	125-4109	#125-4114	125-4114	
T$LAST	= 000001	#5-464	#127-4142	127-4145						
T$LOLI	= 000001	#123-4068	123-4068	#123-4069	123-4069	#125-4109	125-4109	#125-4114	125-4114	
T$LSYM	= 010000	#5-464	5-464	8-565	9-591	74-2776	84-2980	84-2993	89-3208	91-3350
		93-3381	95-3430	97-3468	99-3513	102-3587	104-3614	106-3649	107-3691	109-3723
		111-3754	113-3785	115-3816	117-3938	123-4080	125-4122			
T$LTNO	= 000011	#127-4142								
T$NEST	= 177777	#5-464	5-490	#5-490	5-490	6-516	#6-516	6-516	6-520	6-520
		6-520	#6-520	8-552	#8-552	8-552	8-565	8-565	8-565	#8-565
		9-574	#9-574	9-574	9-591	9-591	9-591	#9-591	9-593	9-593
		9-593	#9-593	10-646	#10-646	10-646	74-2759	#74-2759	74-2759	74-2776
		74-2776	74-2776	#74-2776	84-2976	#84-2976	84-2976	84-2980	84-2980	84-2980
		#84-2980	84-2984	#84-2984	84-2984	84-2993	84-2993	84-2993	#84-2993	88-3105

CREF

SYMBOL CROSS REFERENCE

SYMBOL	VALUE	REFERENCES
		88-3105 88-3105 #88-3105 89-3149 #89-3149 89-3149 89-3156 #89-3156 89-3156
		89-3208 89-3208 89-3208 #89-3208 89-3224 89-3224 89-3224 #89-3224 91-3234
		#91-3234 91-3234 91-3350 91-3350 91-3350 #91-3350 93-3367 #93-3367 93-3367
		93-3381 93-3381 93-3381 #93-3381 95-3404 #95-3404 95-3404 95-3430 95-3430
		95-3430 #95-3430 97-3440 #97-3440 97-3440 97-3468 97-3468 97-3468 #97-3468
		99-3490 #99-3490 99-3490 99-3513 99-3513 99-3513 #99-3513 102-3575 #102-3575
		102-3575 102-3578 #102-3578 102-3578 102-3587 102-3587 102-3587 #102-3587 104-3592
		#104-3592 104-3592 104-3614 104-3614 104-3614 #104-3614 106-3622 #106-3622 106-3622
		106-3649 106-3649 106-3649 #106-3649 107-3657 #107-3657 107-3657 107-3691 107-3691
		107-3691 #107-3691 109-3697 #109-3697 109-3697 109-3723 109-3723 109-3723 #109-3723
		111-3728 #111-3728 111-3728 111-3754 111-3754 111-3754 #111-3754 113-3759 #113-3759
		113-3759 113-3785 113-3785 113-3785 #113-3785 115-3790 #115-3790 115-3790 115-3816
		115-3816 115-3816 #115-3816 117-3821 #117-3821 117-3821 117-3938 117-3938 117-3938
		#117-3938 121-3989 121-3989 121-3989 #121-3989 123-4054 #123-4054 123-4054 123-4065
		#123-4065 123-4065 123-4080 123-4080 123-4080 #123-4080 125-4108 #125-4108 125-4108
		125-4122 125-4122 125-4122 #125-4122 127-4143 127-4143 127-4143 #127-4143
T$NS0	= 000000	#5-490 9-593 #10-646 88-3105 #89-3149 89-3224 #91-3234 91-3350 #93-3367
		93-3381 #95-3404 95-3430 #97-3440 97-3468 #99-3490 99-3513 #102-3575 121-3989
		#123-4054 127-4143
T$NS1	= 000005	#6-516 6-520 #8-552 8-565 #9-574 9-591 #74-2759 74-2776 #84-2976
		84-2980 #84-2984 84-2993 #89-3156 89-3208 #102-3578 102-3587 #104-3592 104-3614
		#106-3622 106-3649 #107-3657 107-3691 #109-3697 109-3723 #111-3728 111-3754 #113-3759
		113-3785 #115-3790 115-3816 #117-3821 117-3938 #123-4065 123-4080 #125-4108 125-4122
T$PCNT	= 000000	#127-4145 127-4146 #127-4146 127-4146
T$PTAB	= 010030	#127-4146 127-4146
T$PTHV	= 000001	5-507 #127-4152
T$PTNU	= 000001	#5-464 127-4146 #127-4146 127-4152 127-4152
T$SAVL	= 177777	#5-464
T$SEGL	= 177777	#5-464
T$SIZE	= 000006	127-4142 #127-4152
T$SUBN	= 000000	#5-464 #102-3578 #104-3592 #106-3622 #107-3657 #109-3697 #111-3728 #113-3759 #115-3790
		#117-3821
T$TAGL	= 177777	#5-464
T$TAGN	= 010032	#5-464 6-516 6-516 #6-516 8-552 8-552 #8-552 9-574 9-574
		#9-574 74-2759 74-2759 #74-2759 84-2976 84-2976 #84-2976 84-2984 84-2984
		#84-2984 89-3156 89-3156 #89-3156 91-3234 91-3234 #91-3234 93-3367 93-3367
		#93-3367 95-3404 95-3404 #95-3404 97-3440 97-3440 #97-3440 99-3490 99-3490
		#99-3490 102-3578 102-3578 #102-3578 104-3592 104-3592 #104-3592 106-3622 106-3622
		#106-3622 107-3657 107-3657 #107-3657 109-3697 109-3697 #109-3697 111-3728 111-3728
		#111-3728 113-3759 113-3759 #113-3759 115-3790 115-3790 #115-3790 117-3821 117-3821
		#117-3821 123-4065 123-4065 #123-4065 125-4108 125-4108 #125-4108 127-4145 127-4145
		#127-4145 127-4146 127-4146 #127-4146 127-4146 127-4146 #127-4146
T$TEMP	= 000000	#6-520 6-520 #7-535 7-535 7-535 #7-535 7-535 7-535 #7-535
		7-535 7-535 #7-535 7-535 7-535 #7-535 7-535 7-535 #7-535
		7-535 7-535 #7-535 7-535 7-535 #7-535 7-535 7-535 #7-535
		7-535 7-535 #7-535 #8-565 8-565 #9-591 9-591 #9-593 9-593
		#74-2776 74-2776 #84-2980 84-2980 #84-2993 84-2993 #88-3105 88-3105 #89-3208
		89-3208 #89-3224 89-3224 #91-3350 91-3350 #93-3381 93 3381 #95-3430 95-3430
		#97-3468 97-3468 #99-3513 99-3513 #102-3580 102-3580 #102-3587 102-3587 #104-3594
		104-3594 #104-3614 104-3614 #106-3624 106-3624 #106-3649 106-3649 #107-3659 107-3659
		#107-3691 107-3691 #109-3699 109-3699 #109-3723 109-3723 #111-3730 111-3730 #111-3754
		111-3754 #113-3761 113-3761 #113-3785 113-3785 #115-3792 115-3792 #115-3816 115-3816

```
SYMBOL  VALUE          REFERENCES
                       #117-3828  117-3828  #117-3938  117-3938  #121-3989  121-3989  #123-4068  123-4068  #123-4068
                        123-4068  #123-4068   123-4068  #12 -4069   123-4069  #123-4069   123-4069  #123-4069   123-4069
                       #123-4070   123-4070  #123-4070   123-4070  #123-4070   123-4070  #123-4071   123-4071  #123-4071
                        123-4071  #123-4071   123-4071  #1 3-4072   123-4072  #123-4072   123-4072  #123-4072   123-4072
                       #123-4080   123-4080  #125-4109   125-4109  #125-4109   125-4109  #125-4109   125-4109  #125-4110
                        125-4110  #125-4110   125-4110  #125-4110   125-4110  #125-4111   125-4111  #125-4111   125-4111
                       #125-4111   125-4111  #125-4112   125-4112  #125-4112   125-4112  #125-4112   125-4112  #125-4113
                        125-4113  #125-4113   125-4113  #125-4113   125-4113  #125-4114   12 -4114  #125-4114   125-4114
                       #125-4114   125-4114  # 5-4115   125-4115  #125-4115   125-4115  #125-4115   125-4115  #125-4122
                        125-4122  #127-4143   127-4143
T$TEST  = 000011       #5-464     102-3578  #102-3578   102-3578   104-3592  #104-3592   104-3592   106-3622  #106-3622
                        106-3622   107-3657  #107-3657   107-3657   109-3697  #109-3697   109-3697   111-3728  #111-3728
                        111-3728   113-3759  #113-3759   113-3759   115-3790  #115-3790   115-3790   117-3821  #117-3821
                        117-3821   127-4142
T$TSTM  = 177777       #5-464      47-1749    49-1836    49-1837    49-1839    52-1960    58-2191    62-2284    68-2512
                        68-2517    68-2520    68-2522    68-2536    68-2543    68-2551    72-2700    72-2714    72-2720
                        72-2727    72-2738    72-2742    72-2746    74-2766    74-2769    74-2772    74-2776    82-2911
                        82-2926    82-2927    82-2929    82-2967    82-2969    84-3001    84-3002    86-3051    86-3052
                        88-3086    88-3091    88-3093    89-3164    89-3166    89-3167    89-3168    89-3169    89-3184
                        89-3185    89-3195    89-3208    91-3242    91-3262    91-3263    91-3269    91-3282    91-3283
                        91-3320    91-3350    93-3369    93-3380    93-3381    93-3389    93-3392    95-3409    95-3430
                        97-3448    97-3468    99-3513   102-3580   102-3587   104-3594   104-3614   106-3624   106-3649
                       107-3659   107-3691   109-3699   109-3723   111-3730   111-3754   113-3761   113-3785   115-3792
                       115-3816   117-3828   117-3918   117-3925   117-3927   117-3929   117-3938
T$TSTS  = 000001       #5-464    #102-3578  #104-3592  #106-3622  #107-3657  #109-3697  #111-3728  #113-3759  #115-3790
                       #117-3821
T$$AU   = 010013       #99-3490    99-3513
T$$AUT  = 010010       #93-3367    93-3381
T$$CLE  = 010011       #95-3404    95-3430
T$$DAT  = 010031       #127-4146   127-4146   127-4151
T$$DU   = 010012       #97-3440    97-3468
T$$HAR  = 010025       #123-4065   123-4065   123-4080
T$$HW   = 010001       #8-552      8-552      8-565
T$$INI  = 010007       #91-3234    91-3350
T$$MSG  = 010003       #74-2759    74-2776
T$$PC   = 000001       #127-4145   127-4152
T$$PRO  = 010000       #6-516
T$$PTA  = 010030       #127-4145   127-4146   #127-4146
T$$RPT  = 010006       #89-3156    89-3208
T$$SOF  = 010026       #125-4108   125-4108    125-4122
T$$SRV  = 010005       #84-2976    84-2980    #84-2984    84-2993
T$$SW   = 010002       #9-574      9-574       9-591
T$$TES  = 010024       #102-3578   102-3580    102-3587  #104-3592   104-3594   104-3614  #106-3622   106-3624   106-3649
                       #107-3657   107-3659    107-3691  #109-3697   109-3699   109-3723  #111-3728   111-3730   111-3754
                       #113-3759   113-3761    113-3785  #115-3790   115-3792   115-3816  #117-3821   117-3828   117-3938
T1             017304  G   7-535  #102-3578
T1TRY   = 000146       G  #17-958
T2             017506  G   7-535  #104-3592
T3             017760  G   7-535  #106-3622
T4             021352  G   7-535  #107-3657
T4TRY   = 000132       G  #17-952
T5             022756  G   7-535  #109-3697
```

SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE		REFERENCES								
T6	023746	G	7-535	#111-3728							
T7	024532	G	7-535	#113-3759							
T8	025522	G	7-535	#115-3790							
T9	026306	G	7-535	#117-3821							
UAM	= 000200	G	#10-653								
UNIT	013362	G	74-2766	#74-2777							
UNITNO	027366		58-2191	*117-3917	117-3925	117-3927	117-3929	*117-3933	#117-3945		
UNREC	012224		68-2551	#68-2576							
UNSUC	011622		68-2516	#68-2533							
UNXPCT	007634		#58-2119								
WAIT	014504		82-2935	82-2943	82-2950	#84-2997	84-3004				
WHCHDR	013640	G	50-1919	50-1933	72-2695	#76-2795					
WRLOCK	= 000026	G	#11-683	70-2651	72-2723						
WRTNO	= 000110	G	#16-931	*50-1935	89-3185	91-3289	91-3291				
WRTN1	= 000112	G	#16-932	*50-1938	89-3195						
XFNSND	006640		50-1886	#50-1891							
XMDB	= 000030	G	#16-911	52-1968	82-2920	84-2979	*91-3306				
XMSR	= 000026	G	#16-910	52-1957	82-2907	82-2913	82-2924	82-2934	82-2942	82-2965	84-2978
			*91-3301								
XSCNT	= 000036	G	#16-914	*102-3582	*104-3600	*106-3631	*106-3631	*106-3631	*107-3666	*107-3666	*107-3666
			*109-3709	*109-3709	*109-3709	*113-3771	*113-3771	*113-3771	*117-3846	*117-3876	*117-3903
XSFLG	= 000034	G	#16-913	50-1904	56-2014	66-2380	*102-3582	*104-3600	*106-3631	*106-3631	*106-3631
			106-3631	106-3632	106-3632	*107-3666	*107-3666	*107-3666	107-3666	107-3667	107-3667
			*109-3709	*109-3709	*109-3709	109-3709	111-3740	111-3740	*113-3771	*113-3771	*113-3771
			113-3771	115-3802	115-3802	*117-3845	*117-3875	*117-3902			
XSPKNM	= 000032	G	#16-912	50-1900	*50-1900	56-2022	58-2096	*58-2100	*58-2113	*58-2119	*58-2135
			*58-2166	66-2379	*102-3582	*104-3600	*106-3631	*106-3631	*106-3631	*106-3631	*106-3632
			*106-3632	*107-3666	*107-3666	*107-3666	*107-3666	*107-3667	*107-3667	*109-3709	*109-3709
			*109-3709	*109-3709	*111-3740	*111-3740	*113-3771	*113-3771	*113-3771	*113-3771	*115-3802
			*115-3802	*117-3847	*117-3877	*117-3904					
XSPTR	= 000106	G	#16-930	*50-1906	58-2101	*58-2102	58-2103	*58-2179			
X$ALWA	= 000000		#5-464								
X$FALS	= 000040		#5-464								
X$OFFS	= 000400		#5-464								
X$TRUE	= 000020		#5-464								

MACRO CROSS REFERENCE
CREF

MACRO NAME	REFERENCES									
BGNAU	#99-3490									
BGNAUT	93-3367									
BGNCLN	#95-3404									
BGNDU	#97-3440									
BGNHRD	123-4065									
BGNHW	#8-552									
BGNINI	91-3234									
BGNMOD	5-490	10-646	89-3149	102-3575	123-4054					
BGNMSG	#74-2759									
BGNPRC	6-516									
BGNPTA	#127-4146									
BGNRPT	89-3156									
BGNSET	#127-4145									
BGNSFT	#125-4108									
BGNSRV	84-2976	84-2984								
BGNSW	#9-574									
BGNTST	102-3578	104-3592	106-3622	107-3657	109-3697	111-3728	113-3759	115-3790	117-3821	
BNCOMP	91-3243	91-3270								
BREAK	47-1749	49-1837	52-1960	62-2284	82-2911	84-3001	84-3002	89-3164	89-3167	89-3169
CLRVEC	82-2967	82-2969	93-3380							
DESCRI	#5-509									
DEVTYP	#21-1027									
DISPAT	7-535									
DOCLN	#49-1839	#91-3263	#91-3283							
DODU	72-2746	93-3392								
DORPT	95-3409									
ENDAU	99-3513									
ENDAUT	#93-3381									
ENDCLN	95-3430									
ENDDU	97-3468									
ENDHRD	#123-4080									
ENDHW	8-565									
ENDINI	#91-3350									
ENDMOD	9-593	88-3105	89-3224	121-3989	127-4143					
ENDMSG	74-2776									
ENDPRO	#6-520									
ENDPTA	127-4151									
ENDRPT	#89-3208									
ENDSET	#127-4152									
ENDSFT	125-4122									
ENDSRV	#84-2980	#84-2993								
ENDSW	9-591									
ENDTST	102-3587	104-3614	106-3649	107-3691	109-3723	111-3754	113-3785	115-3816	117-3938	
EQUALS	10-653									
ERRDF	72-2700	72-2714								
ERRHRD	#72-2727	#72-2742								
ERRSF	#49-1836	#91-3262	#91-3282							
ERRSOF	72-2720	72-2738	86-3051							
EXIT	#102-3580	#104-3594	#106-3624	#107-3659	#109-3699	#111-3730	#113-3761	#115-3792	#117-3828	
GPHARD	91-3269									
GPRMA	#123-4068	#123-4069								
GPRMD	125-4109	125-4114								

```
CZTUUDO    CREATED BY  MACRO  ON 25-FEB-82 AT 07:08     PAGE 22                                                                SEQ 0149
MACRO CROSS REFERENCE                                   CREF
MACRO NAME      REFERENCES
GPRML           #123-4070  #123-4071  #123-4072  #125-4110  #125-4111  #125-4112  #125-4113  #125-4115
HEADER            5-507
LASTAD          127-4142
M$BYTE          #5-507     #5-507     #5-507     #5-507
M$CHEC          #102-3580   102-3580  #104-3594   104-3594  #106-3624   106-3624  #107-3659   107-3659  #109-3699   109-3699
                #111-3730   111-3730  #113-3761   113-3761  #115-3792   115-3792  #117-3828   117-3828
M$CNTO          #123-4068   123-4068  #123-4069   123-4069  #123-4070   123-4070  #123-4071   123-4071  #123-4072   123-4072
                #125-4109   125-4109  #125-4110   125-4110  #125-4111   125-4111  #125-4112   125-4112  #125-4113   125-4113
                #125-4114   125-4114  #125-4115   125-4115
M$COUN          #58-2191    58-2191   #68-2512    68-2512   #68-2517    68-2517   #68-2520    68-2520   #68-2522    68-2522
                #68-2536    68-2536   #68-2543    68-2543   #68-2551    68-2551   #74-2766    74-2766    74-2766    74-2766
                #74-2769    74-2769    74-2769    74-2769    74-2769   #74-2772    74-2772   #86-3052    86-3052   #88-3086
                 88-3086   #88-3091    88-3091   #88-3093    88-3093   #89-3166    89-3166   #89-3168    89-3168   #89-3184
                 89-3184   #89-3185    89-3185    89-3185    89-3185    89-3185    89-3185    89-3185    89-3185    89-3185
                #89-3195    89-3195    89-3195    89-3195    89-3195    89-3195    89-3195    89-3195    89-3195   #93-3389
                 93-3389   #97-3448    97-3448   #117-3918   117-3918  #117-3925   117-3925  #117-3927   117-3927  #117-3929
                 117-3929
M$DATA          #5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                 5-507      5-507      5-507      5-507      5-507      5-507     #5-507      5-507      5-507      5-507
                 5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507
                 5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507      5-507     #5-509
                 5-509     #21-1027    21-1027
M$DECR          #6-520      6-520     #8-565      8-565     #9-591      9-591     #9-593      9-593     #74-2776    74-2776
                #84-2980    84-2980   #84-2993    84-2993   #88-3105    88-3105   #89-3208    89-3208   #89-3224    89-3224
                #91-3350    91-3350   #93-3381    93-3381   #95-3430    95-3430   #97-3468    97-3468   #99-3513    99-3513
                #102-3587   102-3587  #104-3614   104-3614  #106-3649   106-3649  #107-3691   107-3691  #109-3723   109-3723
                #111-3754   111-3754  #113-3785   113-3785  #115-3816   115-3816  #117-3938   117-3938  #121-3989   121-3989
                #123-4080   123-4080  #125-4122   125-4122  #127-4143   127-4143  #127-4146   127-4146
M$DEFA          #123-4068  #123-4068  #123-4069  #123-4069  #123-4070  #123-4070  #123-4071  #123-4071  #123-4072  #123-4072
                #125-4109  #125-4109  #125-4110  #125-4110  #125-4111  #125-4111  #125-4112  #125-4112  #125-4113  #125-4113
                #125-4114  #125-4114  #125-4115  #125-4115
M$ENDE          #8-565     #9-591     #9-593     #74-2776   #84-2980   #84-2993   #88-3105   #89-3208   #89-3224   #91-3350
                #93-3381   #95-3430   #97-3468   #99-3513   #102-3587  #104-3614  #106-3649  #107-3691  #109-3723  #111-3754
                #113-3785  #115-3816  #117-3938  #121-3989  #123-4080  #125-4122  #127-4143
M$ERRI          #49-1836   #49-1836   #72-2700   #72-2700   #72-2714   #72-2714   #72-2720   #72-2720   #72-2727   #72-2727
                #72-2738   #72-2738   #72-2742   #72-2742   #86-3051   #86-3051   #91-3262   #91-3262   #91-3282   #91-3282
M$EXCP          #123-4068   123-4068   123-4068  #123-4069   123-4069   123-4069  #125-4109   125-4109   125-4109  #125-4114
                 125-4114   125-4114
M$EXIT          #102-3580   102-3580  #104-3594   104-3594  #106-3624   106-3624  #107-3659   107-3659  #109-3699   109-3699
                #111-3730   111-3730  #113-3761   113-3761  #115-3792   115-3792  #117-3828   117-3828
M$EXSE          #102-3580  #104-3594  #106-3624  #107-3659  #109-3699  #111-3730  #113-3761  #115-3792  #117-3828
M$EXTJ          #102-3580  #104-3594  #106-3624  #107-3659  #109-3699  #111-3730  #113-3761  #115-3792  #117-3828
M$GEN           #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
                #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
                #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
                #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
                #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
                #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
                #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
                #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-509     #5-509     #6-516     #6-516
                #7-535     #7-535     #8-552     #8-552     #8-552     #8-552     #8-565     #8-565     #9-574     #9-574
                #9-574     #9-574     #9-591     #9-591     #21-1027   #21-1027   #74-2759   #74-2759   #74-2776   #74-2776
```

```
MACRO NAME  REFERENCES
            #84-2976   #84-2976   #84-2980   #84-2980   #84-2984   #84-2984   #84-2993   #84-2993   #89-3156   #89-3156
            #89-3208   #89-3208   #91-3234   #91-3234   #91-3350   #91-3350   #93-3367   #93-3367   #93-3381   #93-3381
            #95-3404   #95-3404   #95-3430   #95-3430   #97-3440   #97-3440   #97-3468   #97-3468   #99-3490   #99-3490
            #99-3513   #99-3513   #102-3578  #102-3578  #102-3587  #102-3587  #104-3592  #104-3592  #104-3614  #104-3614
            #106-3622  #106-3622  #106-3649  #106-3649  #107-3657  #107-3657  #107-3691  #107-3691  #109-3697  #109-3697
            #109-3723  #109-3723  #111-3728  #111-3728  #111-3754  #111-3754  #113-3759  #113-3759  #113-3785  #113-3785
            #115-3790  #115-3790  #115-3816  #115-3816  #117-3821  #117-3821  #117-3938  #117-3938  #123-4065  #123-4065
            #123-4080  #123-4080  #125-4108  #125-4108  #125-4122  #125-4122  #127-4142  #127-4142  #127-4146  #127-4146
            #127-4151  #127-4151
MSGETS      #6-520      6-520     #8-565      8-565     #9-591      9-591     #9-593      9-593     #74-2776    74-2776
            #84-2980    84-2980   #84-2993    84-2993   #88-3105    88-3105   #89-3208    89-3208   #89-3224    89-3224
            #91-3350    91-3350   #93-3381    93-3381   #95-3430    95-3430   #97-3468    97-3468   #99-3513    99-3513
            #102-3587   102-3587  #104-3614   104-3614  #106-3649   106-3649  #107-3691   107-3691  #109-3723   109-3723
            #111-3754   111-3754  #113-3785   113-3785  #115-3816   115-3816  #117-3938   117-3938  #121-3989   121-3989
            #123-4080   123-4080  #125-4122   125-4122  #127-4143   127-4143
MSGETT      #102-3580  #104-3594  #106-3624  #107-3659  #109-3699  #111-3730  #113-3761  #115-3792  #117-3828
MSGNGB      #5-490     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
            #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
            #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
            #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
            #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
            #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
            #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507
            #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-507     #5-509     #5-509     #6-516
            #6-516     #7-535     #7-535     #8-552     #8-552     #8-552     #9-574     #9-574     #9-574     #10-646
            #21-1027   #21-1027   #74-2759   #74-2759   #84-2976   #84-2976   #84-2984   #84-2984   #89-3149   #89-3156
            #89-3156   #91-3234   #91-3234   #93-3367   #93-3367   #95-3404   #95-3404   #97-3440   #97-3440   #99-3490
            #99-3490   #102-3575  #123-4054  #123-4065  #123-4065  #125-4108  #125-4108  #127-4142  #127-4142
MSGNIN      #5-507      5-507      5-507      5-507      5-507      5-507     #5-507      5-507     #5-507      5-507
            #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507
            #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507
            #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507
            #5-507      5-507     #5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507      5-507
            #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507
            #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507
            #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507
            #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507     #5-507      5-507
            #5-509     #5-509      5-509      5-509     #7-535      7-535     #7-535      7-535     #7-535      7-535
            #7-535      7-535     #7-535      7-535     #7-535      7-535     #7-535      7-535     #7-535      7-535
            #7-535      7-535     #7-535      7-535     #8-552      8-552     #9-574      9-574     #21-1027   #21-1027
             21-1027    21-1027   #47-1749    47-1749   #49-1836   #49-1836    49-1836   #49-1836    49-1836   #49-1836
             49-1836   #49-1836    49-1836   #49-1837    49-1837   #49-1839    49-1839   #52-1960    52-1960   #58-2191
            #58-2191    58-2191   #58-2191    58-2191   #58-2191    58-2191    58-2191   #58-2191    58-2191    58-2191
            #62-2284    62-2284   #68-2512   #68-2512    68-2512   #68-2512    68-2512   #68-2512    68-2512    68-2512
            #68-2512    68-2512    68-2512   #68-2517   #68-2517    68-2517   #68-2517    68-2517   #68-2517    68-2517
             68-2517   #68-2517    68-2517    68-2517   #68-2520   #68-2520    68-2520   #68-2520    68-2520    68-2520
            #68-2520    68-2520    68-2520   #68-2522   #68-2522    68-2522   #68-2522    68-2522    68-2522   #68-2522
             68-2522    68-2522   #68-2536   #68-2536    68-2536   #68-2536    68-2536    68-2536   #68-2536    68-2536
             68-2536   #68-2543   #68-2543    68-2543   #68-2543    68-2543   #68-2543    68-2543    68-2543   #68-2543
             68-2543    68-2543   #68-2551   #68-2551    68-2551   #68-2551    68-2551    68-2551   #68-2551    68-2551
             68-2551   #72-2700   #72-2700    72-2700   #72-2700    72-2700   #72-2700    72-2700   #72-2700    72-2700
            #72-2714   #72-2714    72-2714   #72-2714    72-2714   #72-2714    72-2714   #72-2714    72-2714   #72-2720
```

MACRO CROSS REFERENCE

MACRO NAME REFERENCES

#72-2720 72-2720 #72-2720 72-2720 #72-2720 72-2720 #72-2720 72-2720 #72-2727 #72-2727
72-2727 #72-2727 72-2727 #72-2727 72-2727 #72-2727 72-2727 #72-2738 #72-2738 72-2738
#72-2738 72-2738 #72-2738 72-2738 #72-2738 72-2738 #72-2742 #72-2742 72-2742 #72-2742
72-2742 #72-2742 72-2742 #72-2742 72-2742 #72-2746 #72-2746 72-2746 #74-2766 #74-2766
74-2766 74-2766 #74-2766 74-2766 #74-2766 74-2766 74-2766 #74-2766 74-2766 #74-2766
74-2766 74-2766 #74-2766 74-2766 74-2766 #74-2769 #74-2769 74-2769 74-2769 #74-2769
74-2769 74-2769 #74-2769 74-2769 74-2769 #74-2769 74-2769 #74-2769 74-2769 #74-2769
74-2769 74-2769 #74-2769 74-2769 74-2769 #74-2772 #74-2772 74-2772 #74-2772 74-2772
#74-2772 74-2772 74-2772 #74-2772 74-2772 74-2772 #74-2776 74-2776 #82-2911 82-2911
#82-2926 82-2926 #82-2926 82-2926 #82-2927 #82-2927 82-2927 #82-2927 82-2927 #82-2927
82-2927 #82-2927 82-2927 #82-2927 82-2927 82-2927 #82-2929 #82-2929 82-2929 #82-2929
82-2929 #82-2929 82-2929 #82-2929 82-2929 #82-2929 82-2929 #82-2967 82-2967
#82-2967 82-2967 #82-2969 82-2969 #82-2969 82-2969 #84-2980 84-2980 #84-2993 84-2993
#84-3001 84-3001 #84-3002 84-3002 #86-3051 #86-3051 86-3051 #86-3051 86-3051 #86-3051
86-3051 #86-3051 86-3051 #86-3052 #86-3052 86-3052 #86-3052 86-3052 #86-3052 86-3052
86-3052 #86-3052 86-3052 86-3052 #88-3086 #88-3086 88-3086 #88-3086 88-3086 #88-3086
#88-3086 88-3086 88-3086 #88-3086 88-3086 88-3086 #88-3091 #88-3091 88-3091 #88-3091
88-3091 88-3091 #88-3091 88-3091 88-3091 #88-3093 #88-3093 88-3093 #88-3093 88-3093
88-3093 #88-3093 88-3093 88-3093 #89-3164 89-3164 #89-3166 #89-3166 89-3166 #89-3166
89-3166 #89-3166 89-3166 89-3166 89-3166 #89-3167 89-3167 #89-3168 89-3168 89-3168
#89-3168 89-3168 #89-3168 89-3168 89-3168 89-3168 #89-3169 89-3169 #89-3184 #89-3184
89-3184 #89-3184 89-3184 #89-3184 89-3184 89-3184 #89-3184 89-3184 89-3184 #89-3185
#89-3185 89-3185 #89-3185 89-3185 #89-3185 89-3185 #89-3185 89-3185 #89-3185 89-3185
89-3185 #89-3185 89-3185 #89-3185 89-3185 #89-3185 #89-3185 89-3185 #89-3185 #89-3185
89-3185 89-3185 #89-3185 89-3185 89-3185 #89-3195 #89-3195 89-3195 #89-3195 89-3195
#89-3195 89-3195 #89-3195 89-3195 #89-3195 89-3195 89-3195 #89-3195 89-3195 #89-3195
89-3195 #89-3195 89-3195 #89-3195 89-3195 #89-3195 89-3195 89-3195 #89-3195 89-3195
89-3195 #89-3208 89-3208 #91-3242 91-3242 #91-3242 91-3242 #91-3243 91-3243 #91-3262
#91-3262 91-3262 #91-3262 91-3262 #91-3262 91-3262 #91-3262 91-3262 #91-3263 91-3263
#91-3269 91-3269 #91-3269 91-3269 #91-3269 91-3269 #91-3270 91-3270 #91-3282 #91-3282
91-3282 #91-3282 91-3282 #91-3282 91-3282 #91-3282 91-3282 #91-3283 91-3283 #91-3320
91-3320 #91-3320 91-3320 #91-3350 91-3350 #93-3369 #93-3369 93-3369 #93-3369 93-3369
#93-3369 93-3369 #93-3369 93-3369 #93-3369 93-3369 93-3369 #93-3380 93-3380 #93-3380
93-3380 #93-3381 93-3381 #93-3389 #93-3389 93-3389 #93-3389 93-3389 93-3389 #93-3389
93-3389 93-3389 #93-3392 #93-3392 93-3392 #95-3409 95-3409 #95-3430 95-3430 #97-3448
#97-3448 97-3448 #97-3448 97-3448 #97-3448 97-3448 #97-3448 97-3448 97-3448
#97-3468 97-3468 #99-3513 99-3513 #102-3580 102-3580 #102-3580 102-3580 #102-3587 102-3587
#104-3594 104-3594 #104-3594 104-3594 #104-3614 104-3614 #106-3624 106-3624 #106-3624 106-3624
#106-3649 106-3649 #107-3659 107-3659 #107-3659 107-3659 #107-3691 107-3691 #109-3699 109-3699
#109-3699 109-3699 #109-3723 109-3723 #111-3730 111-3730 #111-3730 111-3730 #111-3754 111-3754
#113-3761 113-3761 #113-3761 113-3761 #113-3785 113-3785 #115-3792 115-3792 #115-3792 115-3792
#115-3816 115-3816 #117-3828 117-3828 #117-3828 117-3828 #117-3918 #117-3918 117-3918 #117-3918
117-3918 117-3918 #117-3918 117-3918 117-3918 #117-3925 #117-3925 117-3925 #117-3925 117-3925
#117-3925 117-3925 117-3925 #117-3925 117-3925 117-3925 #117-3927 #117-3927 117-3927 #117-3927
117-3927 #117-3927 117-3927 117-3927 #117-3927 117-3927 117-3927 #117-3929 #117-3929 117-3929
#117-3929 117-3929 #117-3929 117-3929 117-3929 #117-3929 117-3929 117-3929 #117-3938 117-3938
#123-4065 123-4065 #123-4068 123-4068 123-4068 123-4068 123-4068 #123-4069 123-4069 123-4069
123-4069 123-4069 #123-4070 123-4070 123-4070 123-4070 #123-4071 123-4071 123-4071 123-4071
#123-4072 123-4072 123-4072 123-4072 #123-4080 123-4080 #125-4108 125-4108 #125-4109 125-4109
125-4109 125-4109 125-4109 125-4109 #125-4110 125-4110 125-4110 125-4110 #125-4111 125-4111
125-4111 125-4111 #125-4112 125-4112 125-4112 125-4112 #125-4113 125-4113 125-4113 125-4113
#125-4114 125-4114 125-4114 125-4114 125-4114 125-4114 #125-4115 125-4115 125-4115 125-4115

MACRO CROSS REFERENCE

MACRO NAME	REFERENCES									
	#125-4122	125-4122	#127-4142	127-4142	127-4142	127-4142	#127-4146	#127-4146	127-4146	127-4146
MSGNTA	#8-565	#8-565	#9-591	#9-591	#74-2776	#74-2776	#84-2980	#84-2980	#84-2993	#84-2993
	#89-3208	#89-3208	#91-3350	#91-3350	#93-3381	#93-3381	#95-3430	#95-3430	#97-3468	#97-3468
	#99-3513	#99-3513	#102-3587	#102-3587	#104-3614	#104-3614	#106-3649	#106-3649	#107-3691	#107-3691
	#109-3723	#109-3723	#111-3754	#111-3754	#113-3785	#113-3785	#115-3816	#115-3816	#117-3938	#117-3938
	#123-4080	#123-4080	#125-4122	#125-4122	#127-4146	#127-4146	#127-4151	#127-4151		
MSGNTE	#102-3578	#102-3578	#104-3592	#104-3592	#106-3622	#106-3622	#107-3657	#107-3657	#109-3697	#109-3697
	#111-3728	#111-3728	#113-3759	#113-3759	#115-3790	#115-3790	#117-3821	#117-3821		
MSHAPT	#5-507	#5-507								
MSHNAP	#5-507	#5-507								
MSINCR	#5-490	#5-490	#6-516	#6-516	#6-516	#6-516	#8-552	#8-552	#8-552	#8-552
	#9-574	#9-574	#9-574	#9-574	#10-646	#10-646	#47-1749	#49-1836	#49-1837	#49-1839
	#52-1960	#58-2191	#62-2284	#68-2512	#68-2517	#68-2520	#68-2522	#68-2536	#68-2543	#68-2551
	#72-2700	#72-2714	#72-2720	#72-2727	#72-2738	#72-2742	#72-2746	#74-2759	#74-2759	#74-2759
	#74-2759	#74-2766	#74-2769	#74-2772	#74-2776	#82-2911	#82-2926	#82-2927	#82-2929	#82-2967
	#82-2969	#84-2976	#84-2976	#84-2976	#84-2976	#84-2984	#84-2984	#84-2984	#84-2984	#84-3001
	#84-3002	#86-3051	#86-3052	#88-3086	#88-3091	#88-3093	#89-3149	#89-3149	#89-3156	#89-3156
	#89-3156	#89-3156	#89-3164	#89-3166	#89-3167	#89-3168	#89-3169	#89-3184	#89-3185	#89-3195
	#89-3208	#91-3234	#91-3234	#91-3234	#91-3234	#91-3242	#91-3262	#91-3263	#91-3269	#91-3282
	#91-3283	#91-3320	#91-3350	#93-3367	#93-3367	#93-3367	#93-3367	#93-3369	#93-3380	#93-3381
	#93-3389	#93-3392	#95-3404	#95-3404	#95-3404	#95-3404	#95-3409	#95-3430	#97-3440	#97-3440
	#97-3440	#97-3440	#97-3448	#97-3468	#99-3490	#99-3490	#99-3490	#99-3490	#99-3513	#102-3575
	#102-3575	#102-3578	#102-3578	#102-3578	#102-3578	#102-3578	#102-3578	#102-3580	#102-3587	#104-3592
	#104-3592	#104-3592	#104-3592	#104-3592	#104-3592	#104-3594	#104-3614	#106-3622	#106-3622	#106-3622
	#106-3622	#106-3622	#106-3622	#106-3624	#106-3649	#107-3657	#107-3657	#107-3657	#107-3657	#107-3657
	#107-3657	#107-3659	#107-3691	#109-3697	#109-3697	#109-3697	#109-3697	#109-3697	#109-3697	#109-3699
	#109-3723	#111-3728	#111-3728	#111-3728	#111-3728	#111-3728	#111-3728	#111-3730	#111-3754	#113-3759
	#113-3759	#113-3759	#113-3759	#113-3759	#113-3759	#113-3761	#113-3785	#115-3790	#115-3790	#115-3790
	#115-3790	#115-3790	#115-3790	#115-3792	#115-3816	#117-3821	#117-3821	#117-3821	#117-3821	#117-3821
	#117-3821	#117-3828	#117-3918	#117-3925	#117-3927	#117-3929	#117-3938	#123-4054	#123-4054	#123-4065
	#123-4065	#123-4065	#123-4065	#125-4108	#125-4108	#125-4108	#125-4108	#127-4145	#127-4145	#127-4146
	#127-4146	#127-4146	#127-4146							
MSLDRO	#72-2746	72-2746	#82-2926	82-2926	#82-2967	82-2967	#82-2969	82-2969	#91-3242	91-3242
	#91-3269	91-3269	#93-3380	93-3380	#93-3392	93-3392				
MSMCHI	#5-464	#5-464								
MSMCLO	#5-464	#5-464								
MSPOP	#6-520	6-520	#8-565	8-565	#9-591	9-591	#9-593	9-593	#74-2776	74-2776
	#84-2980	84-2980	#84-2993	84-2993	#88-3105	88-3105	#89-3208	89-3208	#89-3224	89-3224
	#91-3350	91-3350	#93-3381	93-3381	#95-3430	95-3430	#97-3468	97-3468	#99-3513	99-3513
	#102-3587	102-3587	#104-3614	104-3614	#106-3649	106-3649	#107-3691	107-3691	#109-3723	109-3723
	#111-3754	111-3754	#113-3785	113-3785	#115-3816	115-3816	#117-3938	117-3938	#121-3989	121-3989
	#123-4080	123-4080	#125-4122	125-4122	#127-4143	127-4143				
MSPRIN	#58-2191	#58-2191	#68-2512	#68-2512	#68-2517	#68-2517	#68-2520	#68-2520	#68-2522	#68-2522
	#68-2536	#68-2536	#68-2543	#68-2543	#68-2551	#68-2551	#74-2766	#74-2766	#74-2769	#74-2769
	#74-2772	#74-2772	#86-3052	#86-3052	#88-3086	#88-3086	#88-3091	#88-3091	#88-3093	#88-3093
	#89-3166	#89-3166	#89-3168	#89-3168	#89-3184	#89-3184	#89-3185	#89-3185	#89-3195	#89-3195
	#93-3389	#93-3389	#97-3448	#97-3448	#117-3918	#117-3918	#117-3925	#117-3925	#117-3927	#117-3927
	#117-3929	#117-3929								
MSPUSH	#5-490	#5-490	#6-516	#6-516	#8-552	#8-552	#9-574	#9-574	#10-646	#10-646
	#74-2759	#74-2759	#84-2976	#84-2976	#84-2984	#84-2984	#89-3149	#89-3149	#89-3156	#89-3156
	#91-3234	#91-3234	#93-3367	#93-3367	#95-3404	#95-3404	#97-3440	#97-3440	#99-3490	#99-3490
	#102-3575	#102-3575	#102-3578	#102-3578	#104-3592	#104-3592	#106-3622	#106-3622	#107-3657	#107-3657

MACRO NAME	REFERENCES									
	#109-3697	#109-3697	#111-3728	#111-3728	#113-3759	#113-3759	#115-3790	#115-3790	#117-3821	#117-3821
	#123-4054	#123-4054	#123-4065	#123-4065	#125-4108	#125-4108				
M$PUT	#58-2191	#58-2191	#58-2191	#58-2191	#68-2512	#68-2512	#68-2512	#68-2512	#68-2517	#68-2517
	#68-2517	#68-2517	#68-2520	#68-2520	#68-2520	#68-2522	#68-2522	#68-2522	#68-2536	#68-2536
	#68-2536	#68-2543	#68-2543	#68-2543	#68-2543	#68-2551	#68-2551	#68-2551	#74-2766	#74-2766
	#74-2766	#74-2766	#74-2766	#74-2766	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769
	#74-2769	#74-2772	#74-2772	#74-2772	#74-2772	#82-2927	#82-2927	#82-2927	#82-2927	#82-2927
	#82-2929	#82-2929	#82-2929	#82-2929	#82-2929	#86-3052	#86-3052	#86-3052	#86-3052	#88-3086
	#88-3086	#88-3086	#88-3086	#88-3091	#88-3091	#88-3091	#88-3093	#88-3093	#88-3093	#89-3166
	#89-3166	#89-3166	#89-3168	#89-3168	#89-3168	#89-3184	#89-3184	#89-3184	#89-3184	#89-3185
	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185
	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195
	#89-3195	#93-3369	#93-3369	#93-3369	#93-3369	#93-3369	#93-3389	#93-3389	#93-3389	#97-3448
	#97-3448	#97-3448	#97-3448	#117-3918	#117-3918	#117-3918	#117-3925	#117-3925	#117-3925	#117-3925
	#117-3927	#117-3927	#117-3927	#117-3927	#117-3929	#117-3929	#117-3929	#117-3929		
M$PUT1	#58-2191	#58-2191	#58-2191	#58-2191	#58-2191	#58-2191	#68-2512	#68-2512	#68-2512	#68-2512
	#68-2512	#68-2512	#68-2517	#68-2517	#68-2517	#68-2517	#68-2517	#68-2517	#68-2520	#68-2520
	#68-2520	#68-2520	#68-2522	#68-2522	#68-2522	#68-2522	#68-2536	#68-2536	#68-2536	#68-2536
	#68-2543	#68-2543	#68-2543	#68-2543	#68-2543	#68-2543	#68-2551	#68-2551	#68-2551	#68-2551
	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766
	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769
	#74-2769	#74-2769	#74-2772	#74-2772	#74-2772	#74-2772	#74-2772	#74-2772	#82-2927	#82-2927
	#82-2927	#82-2927	#82-2927	#82-2927	#82-2927	#82-2927	#82-2929	#82-2929	#82-2929	#82-2929
	#82-2929	#82-2929	#82-2929	#82-2929	#86-3052	#86-3052	#86-3052	#86-3052	#86-3052	#86-3052
	#88-3086	#88-3086	#88-3086	#88-3086	#88-3086	#88-3086	#88-3091	#88-3091	#88-3091	#88-3091
	#88-3093	#88-3093	#88-3093	#88-3093	#89-3166	#89-3166	#89-3166	#89-3166	#89-3168	#89-3168
	#89-3168	#89-3168	#89-3184	#89-3184	#89-3184	#89-3184	#89-3184	#89-3184	#89-3185	#89-3185
	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185
	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3195	#89-3195
	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195
	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#93-3369	#93-3369
	#93-3369	#93-3369	#93-3369	#93-3369	#93-3369	#93-3369	#93-3389	#93-3389	#93-3389	#93-3389
	#97-3448	#97-3448	#97-3448	#97-3448	#97-3448	#97-3448	#117-3918	#117-3918	#117-3918	#117-3918
	#117-3925	#117-3925	#117-3925	#117-3925	#117-3925	#117-3925	#117-3927	#117-3927	#117-3927	#117-3927
	#117-3927	#117-3927	#117-3929	#117-3929	#117-3929	#117-3929	#117-3929	#117-3929		
M$RADI	#123-4068	#123-4068	#123-4069	#123-4069	#123-4070	#123-4070	#123-4071	#123-4071	#123-4072	#123-4072
	#125-4109	#125-4109	#125-4110	#125-4110	#125-4111	#125-4111	#125-4112	#125-4112	#125-4113	#125-4113
	#125-4114	#125-4114	#125-4115	#125-4115						
M$RNRO	#91-3269	#91-3269	#91-3320	#91-3320						
M$SETS	#5-490	#5-490	#6-516	#6-516	#8-552	#8-552	#9-574	#9-574	#10-646	#10-646
	#74-2759	#74-2759	#84-2976	#84-2976	#84-2984	#84-2984	#89-3149	#89-3149	#89-3156	#89-3156
	#91-3234	#91-3234	#93-3367	#93-3367	#95-3404	#95-3404	#97-3440	#97-3440	#99-3490	#99-3490
	#102-3575	#102-3575	#102-3578	#102-3578	#104-3592	#104-3592	#106-3622	#106-3622	#107-3657	#107-3657
	#109-3697	#109-3697	#111-3728	#111-3728	#113-3759	#113-3759	#115-3790	#115-3790	#117-3821	#117-3821
	#123-4054	#123-4054	#123-4065	#123-4065	#125-4108	#125-4108				
M$SVC	#47-1749	47-1749	49-1836	#49-1837	49-1837	#49-1839	49-1839	#52-1960	52-1960	#58-2191
	58-2191	#62-2284	62-2284	#68-2512	68-2512	#68-2517	68-2517	#68-2520	68-2520	#68-2522
	68-2522	#68-2536	68-2536	#68-2543	68-2543	#68-2551	68-2551	72-2700	72-2714	72-2720
	72-2727	72-2738	72-2742	#72-2746	72-2746	#74-2766	74-2766	#74-2769	74-2769	#74-2772
	74-2772	#74-2776	74-2776	#82-2911	82-2911	#82-2926	82-2926	#82-2927	82-2927	#82-2929
	82-2929	#82-2967	82-2967	#82-2969	82-2969	#84-3001	84-3001	#84-3002	84-3002	86-3051
	#86-3052	86-3052	#88-3086	88-3086	#88-3091	88-3091	#88-3093	88-3093	#89-3164	89-3164

MACRO NAME	REFERENCES									
	#89-3166	89-3166	#89-3167	89-3167	#89-3168	89-3168	#89-3169	89-3169	#89-3184	89-3184
	#89-3185	89-3185	#89-3195	89-3195	#89-3208	89-3208	#91-3242	91-3242	91-3262	#91-3263
	91-3263	#91-3269	91-3269	91-3282	#91-3283	91-3283	#91-3320	91-3320	#91-3350	91-3350
	#93-3369	93-3369	#93-3380	93-3380	#93-3381	93-3381	#93-3389	93-3389	#93-3392	93-3392
	#95-3409	95-3409	#95-3430	95-3430	#97-3448	97-3448	#97-3468	97-3468	#99-3513	99-3513
	#102-3580	102-3580	#102-3587	102-3587	#104-3594	104-3594	#104-3614	104-3614	#106-3624	106-3624
	#106-3649	106-3649	#107-3659	107-3659	#107-3691	107-3691	#109-3699	109-3699	#109-3723	109-3723
	#111-3730	111-3730	#111-3754	111-3754	#113-3761	113-3761	#113-3785	113-3785	#115-3792	115-3792
	#115-3816	115-3816	#117-3828	117-3828	#117-3918	117-3918	#117-3925	117-3925	#117-3927	117-3927
	#117-3929	117-3929	#117-3938	117-3938						
M$TLAB	#47-1749	#49-1836	#49-1837	#49-1839	#52-1960	#58-2191	#62-2284	#68-2512	#68-2517	#68-2520
	#68-2522	#68-2536	#68-2543	#68-2551	#72-2700	#72-2714	#72-2720	#72-2727	#72-2738	#72-2742
	#72-2746	#74-2766	#74-2769	#74-2772	#74-2776	#82-2911	#82-2926	#82-2927	#82-2929	#82-2967
	#82-2969	#84-3001	#84-3002	#86-3051	#86-3052	#88-3086	#88-3091	#88-3093	#89-3164	#89-3166
	#89-3167	#89-3168	#89-3169	#89-3184	#89-3185	#89-3195	#89-3208	#91-3242	#91-3262	#91-3263
	#91-3269	#91-3282	#91-3283	#91-3320	#91-3350	#93-3369	#93-3380	#93-3381	#93-3389	#93-3392
	#95-3409	#95-3430	#97-3448	#97-3468	#99-3513	#102-3580	#102-3587	#104-3594	#104-3614	#106-3624
	#106-3649	#107-3659	#107-3691	#109-3699	#109-3723	#111-3730	#111-3754	#113-3761	#113-3785	#115-3792
	#115-3816	#117-3828	#117-3918	#117-3925	#117-3927	#117-3929	#117-3938			
M$TSTL	#47-1749	47-1749	#49-1836	#49-1836	49-1836	#49-1837	49-1837	#49-1839	49-1839	#52-1960
	52-1960	#58-2191	58-2191	#62-2284	62-2284	#68-2512	68-2512	#68-2517	68-2517	#68-2520
	68-2520	#68-2522	68-2522	#68-2536	68-2536	#68-2543	68-2543	#68-2551	68-2551	#72-2700
	#72-2700	72-2700	#72-2714	#72-2714	72-2714	#72-2720	72-2720	72-2720	#72-2727	#72-2727
	72-2727	#72-2738	#72-2738	72-2738	#72-2742	#72-2742	72-2742	#72-2746	72-2746	#74-2766
	74-2766	#74-2769	74-2769	#74-2772	74-2772	#74-2776	74-2776	#82-2911	82-2911	#82-2926
	82-2926	#82-2927	82-2927	#82-2929	82-2929	#82-2967	82-2967	#82-2969	82-2969	#84-3001
	84-3001	#84-3002	84-3002	#86-3051	#86-3051	86-3051	#86-3052	86-3052	#88-3086	88-3086
	#88-3091	88-3091	#88-3093	88-3093	#89-3164	89-3164	#89-3166	89-3166	#89-3167	89-3167
	#89-3168	89-3168	#89-3169	89-3169	#89-3184	89-3184	#89-3185	89-3185	#89-3195	89-3195
	#89-3208	89-3208	#91-3242	91-3242	#91-3262	#91-3262	91-3262	#91-3263	91-3263	#91-3269
	91-3269	#91-3282	#91-3282	91-3282	#91-3283	91-3283	#91-3320	91-3320	#91-3350	91-3350
	#93-3369	93-3369	#93-3380	93-3380	#93-3381	93-3381	#93-3389	93-3389	#93-3392	93-3392
	#95-3409	95-3409	#95-3430	95-3430	#97-3448	97-3448	#97-3468	97-3468	#99-3513	99-3513
	#102-3580	102-3580	#102-3587	102-3587	#104-3594	104-3594	#104-3614	104-3614	#106-3624	106-3624
	#106-3649	106-3649	#107-3659	107-3659	#107-3691	107-3691	#109-3699	109-3699	#109-3723	109-3723
	#111-3730	111-3730	#111-3754	111-3754	#113-3761	113-3761	#113-3785	113-3785	#115-3792	115-3792
	#115-3816	115-3816	#117-3828	117-3828	#117-3918	117-3918	#117-3925	117-3925	#117-3927	117-3927
	#117-3929	117-3929	#117-3938	117-3938						
M$WORD	#5-507	#5-507	#7-535	#7-535	#7-535	#7-535	#7-535	#7-535	#7-535	#7-535
	#7-535	#7-535	#7-535	#49-1836	#49-1836	#49-1836	#49-1836	#72-2700	#72-2700	#72-2700
	#72-2700	#72-2714	#72-2714	#72-2714	#72-2714	#72-2720	#72-2720	#72-2720	#72-2720	#72-2727
	#72-2727	#72-2727	#72-2727	#72-2738	#72-2738	#72-2738	#72-2738	#72-2742	#72-2742	#72-2742
	#72-2742	#86-3051	#86-3051	#86-3051	#86-3051	#91-3262	#91-3262	#91-3262	#91-3262	#91-3282
	#91-3282	#91-3282	#91-3282	#102-3580	#104-3594	#106-3624	#107-3659	#109-3699	#111-3730	#113-3761
	#115-3792	#117-3828	#123-4068	#123-4068	#123-4069	#123-4069	#123-4070	#123-4070	#123-4071	#123-4071
	#123-4072	#123-4072	#125-4109	#125-4109	#125-4110	#125-4110	#125-4111	#125-4111	#125-4112	#125-4112
	#125-4113	#125-4113	#125-4114	#125-4114	#125-4115	#125-4115	#127-4146	#127-4146		
POINTE	5-497									
POP	#23-1083	43-1708	43-1709	52-1961	52-1969	58-2157	60-2221	60-2222	68-2563	68-2564
	72-2747	72-2748	72-2749	72-2750	74-2774	74-2775	78-2840	78-2841	80-2878	80-2879
	86-3056	86-3057	86-3058	88-3094	88-3095	89-3202	89-3203	89-3204	89-3205	89-3206
	89-3207	97-3446	97-3447							

MACRO NAME	REFERENCES									
PRINTB	#68-2536	#74-2766	#74-2769	#74-2772	#86-3052					
PRINTF	58-2191	88-3086	88-3091	88-3093	93-3389	97-3448	117-3918	117-3925	117-3927	117-3929
PRINTS	#89-3166	#89-3168	#89-3184	#89-3185	#89-3195					
PRINTX	68-2512	68-2517	68-2520	68-2522	68-2543	68-2551				
PUSH	#23-1070	43-1701	43-1702	52-1955	52-1959	58-2143	60-2210	60-2211	68-2488	68-2489
	72-2684	72-2685	72-2686	72-2687	74-2760	74-2761	78-2816	78-2817	80-2856	80-2857
	86-3030	86-3031	86-3032	88-3079	88-3080	89-3157	89-3158	89-3159	89-3160	89-3161
	89-3162	97-3442	97-3443							
READEF	#91-3242									
RFLAGS	#91-3320									
SETPRI	82-2926									
SETVEC	82-2927	82-2929	93-3369							
SVC	#5-463	5-464								
SWAPIN	#23-1100	#49-1786	#49-1797	#49-1804	#49-1825					
SWAPOW	#23-1121	50-1882								
TSTID	#35-1525	102-3579	104-3593	106-3623	107-3658	109-3698	111-3729	113-3760	115-3791	
TUREAD	#31-1407	#106-3632	#107-3667	#111-3740	#115-3802					
TURTRY	#29-1332	#106-3631	#106-3632	#107-3666	#107-3667	#109-3709	#111-3740	#113-3771	#115-3802	
TUSEEK	#27-1270	104-3600								
TUSELF	#33-1484	#102-3582								
TUWRIT	#25-1162	106-3631	107-3666	109-3709	113-3771					
XFER	#102-3580	#104-3594	#106-3624	#107-3659	#109-3699	#111-3730	#113-3761	#115-3792	#117-3828	