

TU58

TU58 PERF EXERCISER
CZTUUCO

AH-E649C-MC
FICHE 1 OF 1

APR 1982
COPYRIGHT © 79-82
MADE IN USA



The main body of the document is a large grid of approximately 15 columns and 25 rows. Each cell in the grid contains a small, dense table or data set. The text within these cells is extremely small and difficult to read, but it appears to be organized in a structured format, possibly representing a performance exercise or a data matrix. The overall appearance is that of a microfiche or a high-density data storage format.



.REM 2

IDENTIFICATION

PRODUCT CODE: AC-E648C-MC
PRODUCT NAME: CZTUUC0 TU58 PERF EXER
PRODUCT DATE: SEPT 1981
MAINTAINER: DIAGNOSTIC ENGINEERING GROUP
AUTHOR: R. J. ROSS

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

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

COPYRIGHT (C) 1979,1980,1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

&

.SBTTL REVISION HISTORY

HISTORY

.REM 8

JUNE 18, 1979	INITIAL RELEASE	CZTUUA0
JULY 1, 1979	SECOND RELEASE	CZTUUB0
JUNE 1, 1980	THIRD RELEASE	CZTUUB1
OCTOBER 1, 1981	FOURTH RELEASE	CZTUUC0

CZTUUA0

1. INITIAL REALEASE--PERF. EXER. FOR UP TO 8 TUS8 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 TUS8 INIT SEQUENCE WAS TERMINATED.

CHANGES TO CZTUUB1

1. TEST 8 WAS ADDED TO THE DIAGNOSTICS BECAUSE THE TUS8 HAS BEEN UPDATED TO USE MODIFIED RADIAL SERIAL PROTOCOL.

.REM 8

TABLE OF CONTENTS

- 1.0 GENERAL INFORMATION
- 1.1 PROGRAM ABSTRACT
- 1.2 SYSTEM REQUIREMENTS
- 1.3 RELATED DOCUMENTS AND STANDARDS
- 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
- 1.5 ASSUMPTIONS

- 2.0 OPERATING INSTRUCTIONS
- 2.1 HOW TO RUN THIS DIAGNOSTIC

- 3.0 ERROR INFORMATION

- 4.0 PERFORMANCE AND PROGRESS REPORTS

- 5.0 DEVICE INFORMATION TABLES

- 6.0 TEST SUMMARIES

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 TUS8 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 16K WORDS OF MEMORY AND CONSOLE DEVICE.

TUS8 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 HIERARCHY 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 INVOLVED 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 4-7 -- 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
4-7, THE DRIVE NUMBER (0 OR 1) MAY
BE ADDED TO DATA WRITTEN ON TAPE TO
INSURE DRIVE SELECT BIT OPERATION.

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

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

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

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

PRINT UNIT PROTOCOL SUMMARY (TEST 8) -- PRINTS A TABLE INDICATING
THE PROTOCOL OF EACH UNIT.

3.0 ERROR INFORMATION

ERROR INFORMATION IS PROVIDED ON OCCURRENCE 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:

- OR
1. INIT CODE IN SUPERVISOR IS EXECUTED
 2. INIT IS REQUESTED BY DEVICE AS A RESULT OF ERROR.

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

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

TESTS 4-7: 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 4-7 IS QUITE LONG IF THE FULL TAPE (512.) IS SELECTED.

TEST 4: WRITE TAPE

TEST 5: READ TAPE

TEST 6: 'WRITE VERIFY' TAPE

TEST 7: READ MODIFIED THRESHOLD TAPE

TEST 8: THE FIRST PART OF TEST 8 DETERMINES IF A UNIT IS CAPABLE OF MODIFIED RADIAL SERIAL PROTOCOL. THIS PART OF THE TEST IS WRITTEN USING RADIAL SERIAL PROTOCOL, AND DETERMINES THE PROTOCOL OF A UNIT BY SENDING THE TU58 A GET CHARACTERISTICS COMMAND AND MONITORING THE RESPONSE. IF THE TU58 RETURNS AN END PACKET IT IS A MODIFIED UNIT. IF THE TU58 RETURNS A DATA PACKET IT IS A NON-MODIFIED UNIT. NOTE, THE DATA PACKET RETURNED ON A GET CHARACTERISTICS COMMAND IS NOT NORMAL, RATHER IT CONSISTS OF A DATA PACKET THAT IS 28. BYTES PLUS AN END PACKET WHICH IS 14. BYTES.

THE SECOND PART OF TEST 8 TESTS ONLY THOUGH'S 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 AVAILABLE BY ANSWERING YES (Y) TO SOFTWARE (SW) QUESTION # 5.

&

```

444 .TITLE PROGRAM HEADER AND TABLES
445 .SBTTL PROGRAM HEADER
471
473 .ENABL ABS,AMA
474 002000 = 2000
476 .NLIST BEX
477 002000 BGNMOD
478
479 :++
480 : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
481 : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
482 :--
483
484 002000 POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU,BGNSETUP
485
493
494 002000 HEADER CZTUU,C,0,3600.,1,PRI07

```

```

002000
002000 103
002001 132
002002 124
002003 125
002004 125
002005 000
002006 000
002007 000
002010
002010 103
002011
002011 060
002012 000001
002014 007020
002016 037614
002020 037756
002022 002174
002024 002206
002026 040422
002030 000000
002032 000000
002034 000001
002036 000000
002040 002152
002042 000340
002044

```

```

LSNAME::
        .ASCII /C/
        .ASCII /Z/
        .ASCII /T/
        .ASCII /U/
        .ASCII /U/
        .BYTE 0
        .BYTE 0
        .BYTE 0
LSREV::
        .ASCII /C/
LSDEPO::
        .ASCII /0/
LSUNIT::
        .WORD T$PTHV
L$TIML::
        .WORD 3600.
LSHPCP::
        .WORD L$HARD
L$SPCP:
        .WORD L$SOFT
LSHPTP::
        .WORD L$HW
L$SPTP::
        .WORD L$SW
L$LADP::
        .WORD L$LAST
L$STA::
        .WORD 0
L$CO::
        .WORD 0
L$DTYP::
        .WORD 1
L$APT::
        .WORD 0
L$DTP::
        .WORD L$DISPATCH
L$PRIO::
        .WORD PRI07
L$ENVI::

```

002044 000000
002046
002046 000000
002050
002050 003
002051 003
002052
002052 000000
002054 000000
002056
002056 000000
002060
002060 005506
002062
002062 015014
002064
002064 000000
002066
002066 000000
002070
002070 017146
002072
002072 017022
002074
002074 000000
002076
002076 002122
002100
002100 104035
002102
002102 000000
002104
002104 016030
002106
002106 017002
002110
002110 016620
002112
002112 002142
002114
002114 000000
002116
002116 000000
002120
002120 000000

L\$EXP1:: .WORD 0
L\$MREV:: .WORD 0
L\$EF:: .BYTE C\$REVISION
.BYTE C\$EDIT
L\$SPC:: .WORD 0
L\$DEVP:: .WORD 0
L\$REPP:: .WORD L\$DVTYP
L\$EXP4:: .WORD L\$RPT
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 ESLOAD
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\$DLV:: .WORD 0
L\$HIME:: .WORD 0
L\$DESC:: .ASCIZ /TU58 PERF EXER/
.EVEN

495
496

DESCRIP <TU58 PERF EXER>

002122
002122 124 125 065
002122

```
498  
499  
500  
501  
502  
503 002142  
    002142  
504 002142 000000  
505 002144 177777  
506 002146 177777  
507 002150  
  
    :++  
    :THE PROTECT TABLE IS USED BY THE MONITOR TO WARN THE OPERATOR WHEN HE  
    :TRIES TO TEST THE LOAD DEVICE.  
    :--  
  
    BGNPROT  
  
    .WORD 0  
    .WORD -1  
    .WORD -1  
  
    :DEVICE CSR  
    :NO MASS BUS  
    :NO DRIVE  
  
    LSPROT::  
  
    ENDPROT
```

514
515
516
517
518
519
520
521
522 002150
002150 000010
002152
002152 017150
002154 017352
002156 017624
002160 021230
002162 022220
002164 023004
002166 023774
002170 024560
523

.SBTTL DISPATCH TABLE

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

DISPATCH 8

.WORD 8
LSDISPATCH::
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8

```
531          .SBTTL  DEFAULT HARDWARE P-TABLE
532
533          :++
534          : THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
535          : THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
536          : IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES.
537          :--
538
539          002172          BGNHW  DFPTBL
540          002172          000004
541          002174          176500          .WORD  L10001-LSHW/2
542          002176          000300          LSHW::
543          002200          000003          DFPTBL::
544          002202          000000          .WORD  176500          ;CSR ADDRESS
545          545          .WORD  300          ;VECTOR ADDR.
551          551          .WORD  3          ;TEST DRIVE ZERO AND ONE
552          002204          ENDHW          .WORD  0          ;NOT PDT TYPE INTERFACE
553          002204
```

```
554 .SBTTL SOFTWARE P-TABLE
555
556 :++
557 : THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
558 : PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
559 :--
560
561 002204          BGNSW  SFPTBL
    002204 000007
    002206
    002206
                                LSSW:: .WORD L10002-LSSW/2
                                SFPTBL::
562
563 002206 000010
564 002210 000001
565 002212 000001
566 002214 000001
567 002216 000001
568 002220 000001
569 002222 000000
570
571
572
573
574
575
576
577
578 002224          ENDSW
    002224
                                L10002:
579
580 002224          ENDMOD
```

```
LENGTH: .WORD 8.
STAEOP: .WORD 1
PRBUF: .WORD 1
CMPDAT: .WORD 1
DRVCHK: .WORD 1
EVLTHR: .WORD 1
PPSOT8: .WORD 0
;TAPE LENGTH
;PRINT STATISTICS AT EOP
;PRINT DATA BUF ON COMP. ERROR
;COMPARE DATA
;ADD DR # TO DATA
;THRESHOLD FOR EVL TEST
;PRINT UNIT PROTOCOL SUMMARY (TST8)
```

593
594
622
632
633 002224
634
635
636
637
638
639
640 002224

.TITLE GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD

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

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

```
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
```

641

G
S

```
654          .SBTTL  ERROR CODE EQUATES
655
656          ;THE ERROR CODE OFFSET VALUES :
657          ;USED BY ROUTINE 'LOG' TO INDEX (BY R5) INTO DEVICE'S DATA BLOCK AND
658          ;INCREMENT STATISTICS.
659
660          000002      SFTRD   ==      2
661          000004      SFTWR   ==      4
662          000006      RCINIT  ==      6
663          000010      OTL     ==      8.
664          000012      OVRN    ==     10.
665          000014      BDCOM   ==     12.
666          000016      HRDRD   ==     14.
667          000020      HRDWR   ==     16.
668          000022      BDCHK   ==     18.
669          000024      SKERR   ==     20.
670          000026      WRLOCK  ==     22.
671          000030      NOMOT   ==     24.
672          000032      CNINIT  ==     26.
673          000034      PARTL   ==     28.
674          000036      NOUNIT  ==     30.
675          000040      CMNDER  ==     32.
676          000042      RECERR  ==     34.
677          000044      SLFER   ==     36.
678          000046      SUCOTL  ==     38.
679          000050      TORCVB  ==     40.
680          000054      NCART   ==     44.
681          000056      TOSNDB  ==     46.
682
683          ;          IN ADDITION, SYSTEM SETUP OR RUNTIME ERRORS ARE:
684
685          ;          100.  -      ALL UNITS ABORTED
686
687          ;          101.  -      MORE THAN 8. UNITS (16 DRIVES) REQUESTED
688
689          ;          102.  -      NEITHER DRIVE SELECTED FOR THIS CONTROLLER
690
691          ;          ALL THE ABOVE ARE CLASSIFIED AS SYSTEM FATAL
692
```

```

694 .SBTTL GENERAL EQUATES
695 ;RADIAL SERIAL CODES:
696 -----
697 ;THE FLAG BYTE CODES ARE:
698     RSCMND == 2           ;"COMMAND" PACKET
699     RSCONT == 20          ;"CONTINUE" SINGLE BYTE
700     RSXON  == 20          ;"XON" SINGLE BYTE
701     RSXOFF == 23          ;"XOFF" SINGLE BYTE
702     RSINIT == 4           ;"INIT" SINGLE BYTE
703     RSDATA == 1           ;"DATA" PACKET
704     RSEND  == RSCMND      ;"END" PACKET FLAG IS "COMMAND"
705 -----
706 ;END PACK SIZE:
707     RSNDZ == 14.          ;TOTAL BYTES IN COMMAND PACKET
708 ;MESSAGE PACK SIZE:
709     RSMSZ == 12           ;10. BYTES FOR BYTE COUNT INSIDE CMND PACK
710 ;DATA PACK SIZE:
711     RSDSZ == 132.         ;TOTAL BYTES IN DATA PACKET
712 ;DATA + END PACK SIZE:
713     RSDNSZ == RSDSZ+RSNDZ
714 ;GET CHARACTERISTICS DATA PACKET SIZE
715     RSGCDP == 28.         ;TOTAL BYTES FOR GET CHAR DATA PACKET
716                                     ;MINUS THE END PACKET
717     RSSNSZ = RSMSZ + 4     ;SIZE FOR SENDING COMMAND PACK
718     RCBFSZ == 4*RSDSZ+RSNDZ ;4 DATA PAKS AND END PACK
719                                     ;IS SIZE OF RCV BUFFERS
720 -----
721 ;
722     THE OP CODES ARE:
723     RSSEND == 100          ;END PACK DESCRIPTOR
724     RSSWR  == 3           ;WRITE
725     RSSRD  == 2           ;READ
726     RSSSEK == 5           ;SEEK
727     RSSGET == 12          ;GET CHARACTERISTICS
728     RSSNOP == 0           ;NO-OPERATION
729     RSSNIT == 1           ;INITIALIZE
730     RSSSLF == 7           ;SELF TEST
731 -----
732 ;THE SUCCESS CODES ARE:
733
734     ESABO ==-48.          ;BAD COMMAND FROM HOST
735     ESNCR ==-9.           ;NO CARTRIDGE
736     ESNONX ==-8.          ;NO DRIVE
737     ESOK   ==0            ;OP COMPLETE SUCCESS
738     ESPART ==-2           ;PARTIAL OP
739     ESSK   ==-32.         ;SEEK ERROR
740     ESTRY  ==1            ;RETRY OCCURRED
741     ESWLOC ==-11.         ;WRITE PROTECTED
742     ESNOMO ==-33.         ;MOTOR STOPPED
743     ESCMD  ==-48.         ;COMMAND ERROR
744     ESREC  ==-55.         ;BAD RECORD NUMBER.
745     ESCKS  ==-17.         ;TU CHKSUM ERROR
746     ESSLF  ==-1.          ;SELF TEST ERROR
747     ESCKSM=ESCKS
748     ESWR=ESCKS
749     ESRD=ESCKS
750 -----

```

G S

752
753
754
755
756
757 002224 002320
758 002226 003052
759 002230 003112
760 002232 002534
761 002234 002776
762 002236 003256
763 002240 002402
764 002242 003152
765 002244 003214
766 002246 002554
767 002250 002304
768 002252 002512
769 002254 002444
770 002256 002616
771 002260 002632
772 002262 002654
773 002264 002702
774 002266 002716
775 002270 002362
776 002272 002736
777 002274 002762
778 002276 002320
779 002300 002462
780 002302 003030

.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).

RSNTAB: MSNLOG
MSSFRD
MSSFWR
MSRNIT
MSQRSP
MSOVRN
MSCOM
MSHDRD
MSHDWR
MSHCHK
MSSKER
MSWPRO
MSNOMO
MSNIT
MSPART
MSUNIT
MSCMD
MSREC
MSSELF
MSWRSP
MSNRSP
MSNLOG
MSNOTP
MSTOSN

```

782                                     ;HERE ARE THE MESSAGES PROPER:
783
784 002304    123    105    105  MSSKER:: .ASCIZ /SEEK ERROR/           ;DEVICE COULD NOT READ HEADER
785                                     .EVEN
786 002320    123    131    123  MSNLOG:: .ASCIZ /SYSTEM ERROR/       ;DIAGNOSTIC HUNG. BETTER RE-BOOT
787                                     .EVEN
788 002336    102    101    104  MSBDA:: .ASCIZ /BAD DATA IN PACKET/   ;HOST DATA CHECK FOUND ERROR, DEVICE MAY
789                                     .EVEN                               ;HAVE READ CORRECTLY.
790 002362    123    105    114  MSSELF:: .ASCIZ /SELF TEST ERROR/     ;MICRO DIAGNOSTIC FAILED, BUT DEVICE COULD STILL
791                                     .EVEN                               ;SEND AN END PACKET.
792 002402    102    101    104  MSCOM:: .ASCIZ /BAD DATA W-O DATA CHECK ERR AT TU/ ;PREVIOUS DATA CHECK
793                                     .EVEN                               ;ERROR NOT DUE TO DEVICE READ OPERATION
794 002444    115    117    124  MSNOMO:: .ASCIZ /MOTOR STOPPED/       ;DEVICE COULD NOT GET ANY MEANINGFUL SIGNAL
795                                     .EVEN                               ;FROM TAPE.
796 002462    103    101    122  MSNOTP:: .ASCIZ /CARTRIDGE NOT IN PLACE/ ;NO MEDIA OR BAD SWITCH
797                                     .EVEN
798 002512    127    122    111  MSWPRO:: .ASCIZ /WRITE PROTECTION/     ;CARTRIDGE WRITE PROTECT TAB MISSING OR
799                                     .EVEN                               ;SWITCH BAD
800 002534    122    105    103  MSRNIT:: .ASCIZ /RECIEVING INIT/       ;DEVICE SENT INIT REQUEST
801                                     .EVEN
802 002554    110    117    123  MSHCHK:: .ASCIZ /HOST FOUND PACKET CHECKSUM ERROR/ ;DEVICE SENT PACK WITH
803                                     .EVEN                               ;BAD CHECKSUM
804 002616    103    101    116  MSNIT:: .ASCIZ /CAN'T INIT/           ;DEVICE SENT BYTE OTHER THAN "CONTINUE"
805                                     .EVEN                               ;DURING INITIALIZATION
806 002632    120    101    122  MSPART:: .ASCIZ /PARTIAL OPERATION/    ;END OF MEDIUM ENCOUNTERED
807                                     .EVEN
808 002654    042    116    117  MSUNIT:: .ASCIZ /'NON-EXISTENT' DRIVE/ ;DEVICE RECV'D TOO LARGE DRIVE NUMBER
809                                     .EVEN
810 002702    102    101    104  MSCMD:: .ASCIZ /BAD COMMAND/         ;DEVICE COULD NOT UNDERSTAND HOST
811                                     .EVEN
812 002716    102    101    104  MSREC:: .ASCIZ /BAD RECORD NO./      ;DEVICE RECV'D TOO LARGE A RECORD NUMBER
813                                     .EVEN
814 002736    127    122    117  MSWRSP:: .ASCIZ /WRONG SUCCESS CODE/   ;HOST COULD NOT DECIPHER CODE IN END PACK
815                                     .EVEN
816 002762    116    117    040  MSNRSP:: .ASCIZ /NO RESPONSE/        ;TIME OUT WAITING FOR BYTE IN RCV BUF ON INTERFACE.
817                                     .EVEN
818 002776    111    116    104  MSQRSP:: .ASCIZ \INDECIPHERABLE FLAG BYTE\ ;HOST COULD NOT UNDERSTAND 1ST BYTE OF
819                                     .EVEN                               ;RESPONSE FROM TU AS PROPER PROTOCOL
820 003030    124    111    115  MSTOSN:: .ASCIZ /TIME OUT ON SEND/    ;DLV 'READY' NEVER WENT HIGH
821                                     .EVEN
822 003052    122    105    103  MSSFRD:: .ASCIZ /RECOV. DATA CHECK ERR ON RD OP/ ;TU58 RESPONDED WITH 'DATA-CHECK'
823                                     .EVEN                               ;ERROR ON READ OP. ;HOST RETRY(S) SUCCESSFUL
824 003112    122    105    103  MSSFWR:: .ASCIZ /RECOV. DATA CHECK ERR ON WR OP/ ;SAME BUT WR OR WR VERIFY OPERATION
825                                     .EVEN
826 003152    125    116    122  MSHDRD:: .ASCIZ /UNRECOV. DATA CHECK ERR ON RD OP/ ;TU58 RESPONDED WITH 'DATA-CHECK'
827                                     .EVEN                               ;ERROR ON READ OP. ;RETRIES UNSUCCESSFUL
828 003214    125    116    122  MSHDWR:: .ASCIZ /UNRECOV. DATA CHECK ERR ON WR OP/ ;SAME BUT WR OPERATION
829                                     .EVEN
830 003256    104    114    126  MSOVRN:: .ASCIZ /DLV ERROR IN RECEIVE/ ;DLV ERROR (THE CONTENTS PRINTED OUT)
831                                     .EVEN
    
```


		.SBTTL DATA BLOCK FORMAT		-----	
868					
869					
870					:R5 --> TOP OF 1 OF THE 8 DATA BLOCKS (1 PER UNIT) DURING EXECUTION
871					:@R5 IS THE STATUS WORD CONTAINING:
872					:BIT15 = ABORTED
873					:BIT14 = SEND 'BREAK'
874					:BIT13 = RETRY FLAG BYTE ERROR (DATA PACKS)
875					:BIT12 = TEMP STOR WRITE MACRO
876					:BIT11 = UNIT NOT BEING TESTED
877					:BIT10 = RETRYING DATA ERROR
878					:BIT9 = TU58 CHKSUM ERROR
879					:BIT8 = RD/WR OPERATION
880					:BIT7 = NORMAL/REDUCED THRESHOLD (MACROS)
881					:BIT6 = HOST DATA COMPARE ERROR
882					:BIT5 = WR VERIFY OPERATION
883					:BIT4 = TYPE OF PAK SENT ODATA 1CMD
884					:BIT3 = RETRY FLAG BYTE ERR.(SEND COMMAND PACK)
885					:BIT0,1,2=UNIT NO.
886	000000	STATUS	==	0.	:DEVICE STATE
887	000002	RETRY	==	2.	:# OF RETRIES
888	000004	ABNDX	==	4.	:ERROR NUMBER FOR LOG
889		:R0			:STORAGE FOR REGISTERS USED IN TEST BODY
890		:R1			:STORED WITH SWAPOW
891		:R2			:RETRIEVED WITH SWAPIN
892		:R3			:
893		:R4			:
894	000020	TSTPC	==	16.	: POINTER TO NEXT EXECUTABLE TEST INST.
895	000022	RCSR	==	18.	:DLV RCV STATUS ADDRESS
896	000024	RCDB	==	20.	:DLV RCV DATA ADDRESS
897	000026	XMSR	==	22.	:DLV SND STATUS ADDRESS
898	000030	XMDB	==	24.	:DLV SND DATA ADDRESS
899	000032	XSPKMM	==	26.	:THE NUMBER OF PACKETS TO RECEIVE
900	000034	XSFLG	==	28.	:THE EXPECTED FLAG OF 1ST PACKET
901	000036	XSCNT	==	30.	:THE EXPECTED COUNT OF 1ST PACKET
902					:FOR MULTIPLE PACKET RECIEVES (MAX.4)
903		:			:CONSECUTIVE XSFLGS AND XSCNTS
904	000060	DR	==	48.	:DR==0 OR 1; BIT8,9 DRIVE SELECTED BY OPERATOR
905	000062	TRK	==	50.	:COUNTER FOR TRACK NUMBER
906	000064	REC	==	52.	:RECORD (BLOCK #)
907					
908	000066	TMP	==	54.	:TEST MACRO REGISTER
909	000070	SNDCNT	==	56.	:THE # OF BYTES FOR SENDING PACKET
910	000072	PATTEN	==	58.	:DATA PATTERN-LOWER BYTE USED
911	000074	DLV	==	60.	:CONTENTS OF RCDB ON DLV ERROR
912	000076	SUCCS	==	62.	:SUCCESS CODE OF LAST END PACKET
913	000100	CMSNT	==	64.	:TYPE OF COMMAND CURRENT IN EVEN BYTE; BIT15==VERIFY OP.
914					
915	000102	RCVBUF	==	66.	: POINTER TO 542. BYTE BUFFER (4 DATA PAKS + END PACK)
916	000104	PKPTR	==	68.	: POINTER TO TOP OF PACKET
917	000106	XSPTR	==	70.	: POINTER TO CURRENTLY USED XSFLG OR XSCNT
918	000110	WRTNO	==	72.	:THE # OF 512. BYTE BLOCKS WRITTEN DR0
919	000112	WRTN1	==	74.	:THE # OF 512. BYTE BLOCKS WRITTEN DR1
920	000114	RDNO	==	76.	:THE # OF 512. BYTE BLOCKS READ DR0
921	000116	RDN1	==	78.	:THE # OF 512. BYTE BLOCKS READ DR1

```

923      ;AND THE ERROR LOG...          +-----+
924      ;SPLIT INTO A BYTE PER DRIVE:  ! DR1 ! DR0 !
925      ;                                +-----+
926
927      ;-----+-----+-----+-----+
928      ;OFFSET IN DATA BLOCK          ;ERROR TYPE      ;ERRCODE;MSG CODE;SUC. CODE
929      ;-----+-----+-----+-----+
930
931      000120  LGOFST ==          80.      ;**RESERVED**
932      000122  SOFTR  ==          82.      ;SOFT READ      ;SFTRD  ;MSSF RD ;ESCKSM
933      000124  SOFTW  ==          84.      ;SOFT WRITE     ;SFTWR  ;MSSFWR ;ESSKSM
934      ;          WORD      ;RECEIVED INIT ;RCINIT ;MSRNIT ;*****
935      ;          WORD      ;BAD FLAG BYTE ;OTL    ;MSQRSP ;*****
936
937      ;THEN THOSE CODES WHICH HAVE N TRIES BEFORE ABORT
938
939      000132  T4TRY  ==          90.      ;DLV ERROR      ;OVRN   ;MSOVRN ;*****
940      000134  BDATA ==          92.      ;BAD DATA      ;BDCOM  ;MSDATA ;*****
941      000136  HARDR  ==          94.      ;HARD READ      ;HRDRD  ;MSHDRD ;ESCKSM
942      000140  HARDW  ==          96.      ;HARD WRITE     ;HRDWR  ;MSHDWR ;ESCKSM
943      ;          WORD      ;CHKSM AT HOST ;BDCHK  ;MSHCHK ;*****
944      ;          WORD      ;SEEK ERROR TOTAL;SKERR  ;MSSKER ;*****
945      000146  T1TRY  ==          102.     ;WRITE PROTECT ;WRLOCK ;MSWPRO ;ESWLOC
946      ;          WORD      ;NO MOTOR       ;NOMOT  ;MSNOMO ;ESNOMO
947      ;          WORD      ;CANT INIT     ;CNINIT ;MSNIT  ;*****
948      ;          WORD      ;PARTIAL OP    ;PARTL  ;MSPART ;ESPART
949      ;          WORD      ;NO UNIT       ;NOUNIT ;MSUNIT ;ESNONX
950      ;          WORD      ;COMMAND ERROR ;CMNDER ;MSCMD  ;ESCMD
951      ;          WORD      ;BAD RECORD NO.;RECERR ;MSREC  ;ESREC
952      ;          WORD      ;SELF TEST ERROR;SLFER  ;MSSELF ;*****
953      ;          WORD      ;WRONG SUC.CODE ;SUCOTL ;MSWRSP ;*****
954      ;          WORD      ;NO RESPONSE   ;TORCVB ;MSNRSP ;*****
955      ;          WORD      ;**RESERVED**
956      ;          WORD      ;NO CARTRIDGE  ;NOCART ;MSNOTP ;ESNCRT
957      ;          WORD      ;TIME OUT SEND;TOSNDB ;MSTOSN ;*****
958
959
960      000202  BLKEND ==          130.     ;OFFSET OF END OF STATISTICS (RESERVED)
961      000204  TUVECT ==          132.     ;VECTOR ADDRESS
962      000206  SAVCNT ==          134.     ;BYTE COUNT SAVED DURING RETRY ON WRITE OPERATION
963      000210  MRSP   ==          136.     ;***** FLAG INDICATING MRSP
964      000212  BLKSIZ ==          138.     ;** RESERVED **
965
;-----+-----+-----+-----+

```

```
968          .SBTTL  DEVICE DATA BLOCK ALLOCATION
969
970
971          ;TABLE OF DEVICE DATA BLOCK ADDRESSES
972
973
974 003346 003366  BLKTBL::      .WORD  DEVO
975 003350 003600          .WORD  DEV1
976 003352 004012          .WORD  DEV2
977 003354 004224          .WORD  DEV3
978 003356 004436          .WORD  DEV4
979 003360 004650          .WORD  DEV5
980 003362 005062          .WORD  DEV6
981 003364 005274  LSTDEV::      .WORD  DEV7
982
983
984          ;AND STORAGE FOR EACH:
985
986 003366  DEVO:      .BLKB  BLKSIZ
987 003600  DEV1:      .BLKB  BLKSIZ
988 004012  DEV2:      .BLKB  BLKSIZ
989 004224  DEV3:      .BLKB  BLKSIZ
990 004436  DEV4:      .BLKB  BLKSIZ
991 004650  DEV5:      .BLKB  BLKSIZ
992 005062  DEV6:      .BLKB  BLKSIZ
993 005274  DEV7:      .BLKB  BLKSIZ
```

```
1009          .SBTTL GLOBAL TEXT SECTION
1010
1011          :
1012          : NAMES OF DEVICES SUPPORTED BY PROGRAM
1013          :
1014 005506          DEVTYP <TUS8 CONTROLLER>
      005506          124      125      065
      005506
1015
1027
1028
1046
```

```
LSDVTYP::
          .ASCIZ /TUS8 CONTROLLER/
          .EVEN
```

```
1055      .SBTTL SYSTEM MACRO DEFINITIONS
1056
1057      .MACRO PUSH ,REG
1058
1059      .NLIST
1060      .LIST ME
1061      .LIST
1062
1063      MOV      REG,-(SP)
1064
1065      .NLIST
1066      .NLIST ME
1067      .LIST
1068      .ENDM
1069
1070      .MACRO POP,REG
1071
1072      .NLIST
1073      .LIST ME
1074      .LIST
1075
1076      MOV      (SP)+,REG
1077
1078      .NLIST
1079      .NLIST ME
1080      .LIST
1081      .ENDM
1082
1083      :++
1084      :THE MACRO 'SWAPIN' RETRIEVES THE TEST REGISTERS WHICH WERE SAVED
1085      :IN THE DEVICE DATA BLOCK.
1086      :--
1087
1088      .MACRO SWAPIN
1089
1090      .NLIST
1091      .LIST ME
1092      .LIST
1093
1094      MOV      6.(R5),R0
1095      MOV      8.(R5),R1
1096      MOV      10.(R5),R2
1097      MOV      12.(R5),R3
1098      MOV      14.(R5),R4
1099
1100      .NLIST
1101      .NLIST ME
1102      .LIST
1103      .ENDM
1104
1105      :++
1106      :THE MACRO 'SWAPOW' SAVES THE CURRENT STATE OF THE UNIT IN THE DRIVE
1107      :DATA BLOCK IN SO THAT THE SCHEDULER MAY 'SWAPIN' ANOTHER UNIT.
1108      :--
1109
1110      .MACRO SWAPOW
1111
1112      .NLIST
1113      .LIST ME
```

```
1112          .LIST
1113
1114          MOV      R0,6.(R5)
1115          MOV      R1,8.(R5)
1116          MOV      R2,10.(R5)
1117          MOV      R3,12.(R5)
1118          MOV      P4,14.(R5)
1119
1120          .NLIST
1121          .NLIST ME
1122          .LIST
          .ENDM
```

1125
 1126
 1127
 1128
 1129
 1130
 1131
 1132
 1133
 1134
 1135
 1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181

```

:++
: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
:(XSPKNT) 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
:          XSPKNT = # OF PACKETS EXPECTED
:          XSFLG = FLAG BYT. 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    #RSCMD,@R0     ;COMMAND FLAG
        MOVB    #RMSIZ,1(R0)   ;THIS SIZE
        MOVB    #RSSWR,2(R0)   ;INSERT OP CODE-WRITE
        MOVB    VER,3.(R0)     ;VERIFY (1 OR 0)
        MOVB    DR,4.(R0)      ;DRIVE #
        MOVB    #O20,5.(R0)    ;MAINTENANCE MODE SWITCH
        CLR     6.(R0)         ;NO SEQUENCE #
        MOV     BCNT,8.(R0)     ;TOTAL COUNT TO WRITE
        MOV     REC,10.(R0)    ;AT RECORD N
        MOV     #RMSIZ,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     #RSCNT,XSFLG(R5) ;THE FLAG
        MOV     #1,XSCNT(R5)   ;THE COUNT
        MOV     #1,XSPKNT(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.
    
```

```

1182          BIS      #BIT12,@R5      ;SO LAST PACKET NOW
1183          BR       C                ;USE REMAINING COUNT
1184          B:      MOV      #128.,R1   ;USE 128. BYTES
1185          C:      MOVB   R1,1(R0)     ;COPY COUNT TO BUFFER
1186          MOV      R1,R3             ;R3=COUNTER TO LOAD BUFFER
1187          MOVB   #RSDATA,@R0        ;FLAG FIRST
1188          TST     (R0)+              ;SKIP COUNT
1189          D:      MOVB   PTRN,(R0)+   ;INSERT DATA
1190          DEC     R3                 ;MORE?
1191          BHI     D                  ;YES
1192          MOV     #TRBUF,R0          ;-->TOP AGAIN
1193          MOVB   1(R0),R1           ;GET COUNT
1194          BIC     #177400,R1        ;ZERO SIGN EXTEND
1195          MOV     R1,SNDCNT(R5)      ;HOW MANY TO SEND PLUS
1196          ADD     #4,SNDCNT(R5)     ;FLAG,COUNT,CHKSUM
1197          ADD     #2,R1              ;COMPENSATE FOR FLAG + COUNT
1198          CALL   CHKSUM             ;FOR CHECKSUM CALC.
1199          MOVB   R1,(R0)+          ;CHKSUM INTO PACKET
1200          SWAB   R1                 ;EVEN ON AN ODD
1201          MOVB   R1,(R0)+          ;BYTE BOUNDARY
1202          BIT     #BIT12,@R5        ;LAST DATA PACKET?
1203          BEQ    E                  ;NO
1204          MOV     #RSEND,XSFLG(R5)  ;YES-EXPECT 'END'
1205          MOV     #RSNDSZ,XSCNT(R5) ;OF THIS SIZE
1206          MOV     #1,XSPKNT(R5)    ;AND 1 PACKET
1207          BR     F                  ;SEND
1208          E:      MOV     #RSCONT,XSFLG(R5) ;(NOT LAST), EXPECT 'CONTINUE'
1209          MOV     #1,XSCNT(R5)      ;AND 1 BYTE
1210          MOV     #1,XSPKNT(R5)    ;AND 1 PACKET
1211          F:      CALL   RSVP        ;SEND PACKET
1212          ;AND RETURN TO SCHEDULER
1213          BIT     #BIT3,@R5         ;FLAG BYTE RETRY?
1214          BNE    T                  ;YES
1215          BIT     #BIT10,@R5        ;RETRY DATA ERROR?
1216          BNE    G                  ;YES
1217          SUB     #128.,R2          ;NO, MORE DATA TO SEND?
1218          BHI    A                  ;YES
1219          BR     H                  ;NO
1220          G:      TURTRY  REC,BCNT,DR  ;RETRY HERE
1221          BIT     #BIT10!BIT3,@R5   ;RETRY AGAIN?
1222          BNE    G                  ;YES
1223          H:      NOP                ;DONE
1224
1225          .NLIST
1226          .NLIST ME
1227          .LIST
1228          .ENDM
1229

```

1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288

```

:++
: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
:         ***
:         *   SUBSEQUENT XSFLGS
:         *   >
:         *   AND XSCNTS
:         ***
:--
    
```

.MACRO TUSEEK REC,DR,?A

.NLIST
 .LIST ME
 .LIST

```

A:      MOV      #TRBUF,R0      ;-->(POINT TO) XMIT BUFFER
        MOV     #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
        CALL   CHKSUM          ;FOR CHECKSUM CALC
        MOV    R1,(R0)         ;RO-->TOP R1=# OF BYTES
        ;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
    
```

1289
1290
1291
1292

.NLIST
.NLIST ME
.LIST
.ENDM

1295
 1296
 1297
 1298
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1330
 1331
 1332
 1333
 1334
 1335
 1336
 1337
 1338
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 1350
 1351

```

:++
:THE RETRY MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO INITIATE
:A RETRY (READ OPERATION) SEQUENCE.
    
```

```

:SETS UP THE EXPECTED PROTOCOL RESPONSES: THE NUMBER OF PACKETS
:(XSPKNM) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT). CALLS
:'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
:CHECKSUM.
    
```

```

: INPUTS - DEVICE BLOCK @R5
:         TRBUF - BUFFER ADDRESS
:         UNITS TEST REGISTERS FROM SWAPIN
    
```

```

: OUTPUTS - SNDCNT(R5) = # OF BYTES TO SEND
:          XSPKNM = # OF PACKETS EXPECTED
:          XSFLG = FLAG BYTE OF 1ST PACKET
:          XSCNT = BYTE COUNT OF 1ST PACKET
    
```

```

:   ***
:   *   SUBSEQUENT XSFLGS
:   *   >
:   *   AND XSCNTS
:   ***
    
```

```

.MACRO TURTRY REC,BCNT,DR,?A,?B,?C,?D,?E
    
```

```

.NLIST
.LIST ME
.LIST
    
```

```

D:   MOV   #TRBUF,R0           ;FORM CMND PACK:
      MOVB  #RSCMND,@R0       ;MESSAGE PACK TYPE
      MOVB  #RSMSIZ,1(R0)     ;THIS BIG
      MOVB  #RSSRD,2(R0)     ;OP CODE-READ
      MOV   REC,10.(R0)       ;THIS RECORD
      MOVB  DR,4.(R0)        ;THIS DRIVE
      CLRB  3(R0)            ;PRESET NORM THRESHOLD
      TSTB  @R5              ;REDUCED?
      BPL   E                 ;NO
      INCB  3(R0)            ;YES-CHANGE THRESHOLD
E:   MOV   BCNT,8.(R0)        ;# BYTES DESIRED
      MOVB  #020,5.(R0)      ;MAINTENANCE MODE
      CLR   6.(R0)           ;NO SEQUENCE #
      MOV   #RSMSIZ,R1       ;SIZE OF PACKET
      TST   (R1)+            ;PLUS FLAG+COUNT INTO R1
      MOV   #RSSNSZ,SNDCNT(R5) ;SET UP SIZE TO SEND

      CALL  CHKSUM           ;FORM CHECKSUM R1=COUNT
      MOV   R1,(R0)         ;INSERT IN PACKET

      MOV   BCNT,R1         ;SET EXPECTATIONS:
                              ;CALC # OF DATA PACKETS TO EXPECT
      MOV   #XSFLG,R3       ;OFFSET OF FLAG
      ADD   R5,R3           ;ABS. ADDR. OF XSFLG
      CLR   R2              ;PRESET
      INC   R2              ;# PACKETS EXPECTED
A:   MOV   #RSDATA,(R3)+    ;LOAD XSFLG
    
```

```
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367
```

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

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

1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426

```

:++
:THE READ MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO INITIATE
:A READ SEQUENCE.
    
```

```

:SETS UP THE EXPECTED PROTOCOL RESPONSES: THE NUMBER OF PACKETS
:(XSPKMM) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT). CALLS
:'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
:CHECKSUM.
    
```

```

: INPUTS - DEVICE BLOCK @R5
:          TRBUF - BUFFER ADDRESS
:          UNITS TEST REGISTERS FROM SWAPIN
    
```

```

: OUTPUTS - SNDCNT(R5) = # OF BYTES TO SEND
:           XSPKMM = # OF PACKETS EXPECTED
:           XSFLG = FLAG BYTE OF 1ST PACKET
:           XSCNT = BYTE COUNT OF 1ST PACKET
:             ***
:             * SUBSEQUENT XSFLGS
:             * >
:             * AND XSCNTS
:             ***
:--
    
```

```

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

```

.NLIST
.LIST ME
.LIST
    
```

```

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

        MOV      BCNT,R1       ;SET EXPECTATIONS:
                                ;CALC # OF DATA PACKETS TO EXPECT:
A:      MOV      #XSFLG,R3      ;GET OFFSET
        ADD      R5,R3          ;ABS. ADDR. OF XSFLG
        CLR      R2             ;PRESET AS NONE
        INC      R2             ;# PACKETS EXPECTED
        MOV      #RSDATA,(R3)+  ;LOAD XSFLG
        MOV      #132.,(R3)+    ;AND EXPECTED COUNT
        SUB      #128.,R1       ;NEG RESULT LAST TIME
        BLOS     C              ;LAST TIME
    
```

1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444

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

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

1447
 1448
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1500
 1501
 1502

```

:++
:THE SELF TEST MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO
:INITIATE A 'DIAGNOSE' SEQUENCE.
:
:SETS UP THE EXPECTED PROTOCOL RESPONSES: THE NUMBER OF PACKETS
:(XSPKMM) 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
:           XSPKMM = # 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,XSPKMM(R5) ;AND 1 PACKET
     ;SEND
     CALL  RSVP          ;RETURN TO SCHEDULER
     BIT   #BIT3,@R5    ;RETRY?(BAD FLAG)
     BNE   A             ;YES
    
```

.NLIST
 .NLIST ME
 .LIST
 .ENDM

```
1505      :++  
1506      :THE TEST ID MACRO INTERFACES THE SUPERVISOR'S TEST DISPATCH TO THE  
1507      :DIAGNOSTIC'S FORMAT BY IMPLEMENTING CALLS THAT: 1) INITIALIZE THE  
1508      :PC OF THE TEST CODE (TSTPC(R5)), 2) ASSIGN THE 1ST DRIVES, 3) RUN  
1509      :THE TEST, 4) SWITCH DRIVES AND REINITIALIZE, 5) RUN THE TEST AGAIN.  
1510      :--  
1511  
1512      .MACRO TSTID  ADDR,?A  
1513  
1514      .NLIST  
1515      .LIST ME  
1516      .LIST  
1517  
1518      MOV      ADDR,TSTTOP      :SAVE ADDR OF TEST  
1519      CALL     SETUP            :INIT UNITS TSTPC  
1520      CALL     SETDR           :GET 1ST DRVS.  
1521      CALL     RUN             :DO TEST  
1522      CALL     SWAPDR          :GET NEXT DRVS.  
1523      BCC      A               :BR NO 2ND DRVS  
1524      CALL     SETUP            :REINIT UNITS TSTPC  
1525      CALL     RUN             :REPEAT TEST  
1526      A:                               :DONE  
1527      .NLIST  
1528      .LIST ME  
1529      .LIST  
1530      .ENDM  
-----
```

```

1533      .SBTTL GLOBAL SUBROUTINES SECTION
1534
1535      :++
1536      : THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES THAT ARE USED
1537      : TO LINK THE DIAGNOSTIC TO THE SUPERVISOR (THROUGH THE TSTID MACRO).
1538      :--
1539
1540      :++
1541      : SWAPDR
1542      : SUBROUTINE TO DETERMINE IF TO TEST OTHER DRIVE (FOR ALL UNITS)
1543      : INPUTS: DR(R5) - DRIVE CONFIGURATION
1544      :          BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
1545      :          LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
1546      :
1547      : OUTPUTS: DR(R5) UPDATED TO TEST SAME OR OTHER DRIVE
1548      :          CARRY SET IF SECOND PASS NECESSARY
1549      :--
1550
1609
1610 005526 005002 SWAPDR: CLR R2 ;FOR # OF DRIVE 1'S.
1611 005530 012737 003346 005626 MOV #BLKTBL,SWPTR ;TABLE ADDR. OF 1ST UNIT
1612 005536 017705 000064 1$: MOV @SWPTR,R5 ;GET DATA BLOCK ADDR.
1613 005542 032715 100000 BIT #BIT15,@R5 ;ABORTED?
1614 005546 001013 BNE 3$ ;YES
1615 005550 032765 000001 000060 BIT #BIT0,DR(R5) ;DID DR. 0?
1616 005556 001007 BNE 3$ ;NO, DID DR.1 1ST PASS
1617 005560 032765 001000 000060 BIT #BIT9,DR(R5) ;YES; 1 SELECTED?
1618 005566 001403 BEQ 3$ ;NO, ALL DONE
1619 005570 105265 000060 INCB DR(R5) ;YES, SWAP
1620 005574 005202 INC R2 ;ONE MORE TO TEST
1621 005576 023727 005626 003364 3$: CMP SWPTR,#LSTDEV ;LAST DEVICE?
1622 005604 103004 BHIS 4$ ;YES
1623 005606 062737 000002 005626 ADD #2,SWPTR ;NO-POINT NEXT
1624 005614 000750 BR 1$ ;DO
1625
1626 005616 005702 4$: TST R2 ;(CLEAR CARRY),MORE TO DO?
1627 005620 001401 BEQ 5$ ;NO
1628 005622 000261 SEC ;YES
1629 005624 000207 5$: RETURN ;RETURN
1630
1631 005626 000000 SWPTR: .WORD
    
```

```

1634
1635      :++
1636      : SETDR - SUBROUTINE TO GET DRIVE FOR 1ST PASS FOR EACH TEST
1637      :
1638      : INPUTS:      DR(R5) - DRIVE CONFIGURATION
1639      :              BLKTBL - TOP OF DATA BLOCK ALLOCAT!ON TABLE
1640      :              LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
1641      :
1642      : OUTPUTS:    DR(R5) IS SET TO TEST DRIVE 0 OR DRIVE 1
1643      :--
1644
1645 005630 012737 003346 005704 SETDR:: MOV    #BLKTBL,SETPTR ;TABLE OF ADDR. 1ST UNIT
1646 005636 017705 000042 1$:    MOV    @SETPTR,R5 ;GET DATA BLOCK ADDR.
1647 005642 105065 000060      CLR    DR(R5) ;PRESET AS DRO
1648 005646 032765 000400 000060      BIT    #BIT8,DR(R5) ;DO DRO?
1649 005654 001002      BNE    2$ ;YES
1650 005656 105265 000060      INCB   DR(R5) ;NO-USE DRIVE 1
1651 005662 023727 005704 003364 2$:    CMP    SETPTR,#LSTDEV ;MORE UNITS
1652 005670 103004      BHS    3$ ;NO-EXIT
1653 005672 062737 000002 005704      ADD    #2,SETPTR ;YES-GET TABLE ENTRY
1654 005700 000756      BR     1$ ;CONFIGURE THAT UNIT
1655 005702 000207      3$:    RETURN
1656 005704 000000      SETPTR: .WORD
    
```

```
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670 005706 012737 003346 006000 CLRALL:: MOV #BLKTBL,CLRPTR ;TOP OF TABLE OF ADDRESSES  
1671 005714 017705 000060 1$: MOV @CLRPTR,R5 ;GET DATA BLOCK  
1672 005720 004737 005746 CALL CLRBUF ;CLEAR IT'S RECEIVE BUFFER  
1673 005724 023727 006000 003364 CMP CLRPTR,#LSTDEV ;LAST DEV?  
1674 005732 103004 BHS 2$ ;YES  
1675 005734 062737 000002 006000 ADD #2,CLRPTR ;-->NEXT  
1676 005742 000764 BR 1$ ;CONTINUE  
1677 005744 000207 2$: RETURN
```

```
:++  
: 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  
:--
```

```

1680
1681      :++
1682      : CLRBUF - CLEARS 1 UNIT'S INPUT BUFFER.
1683      : INPUTS: RCVBUF(R5) IS BUFFER START
1684      :          RCBFSZ - SIZE OF RECEIVE BUFFER IN BYTES
1685      :          RCBFSZ IS SIZE OF BUFFER
1686      : OUTPUTS: CLEARED AREA.
1687      :--
1688 005746 010046 CLRBUF:: PUSH  R0          ;SAVE R0
                                MOV   R0,-(SP)

1689 005750 010446          PUSH  R4          ;SAVE R4
                                MOV   R4,-(SP)

1690 005752 016500 000102      MOV   RCVBUF(R5),R0      ;GET ADDRESS OF BUFFER
1691 005756 012704 001036      MOV   #RCBFSZ,R4        ;SIZE IN BYTES
1692 005762 005020 1$:      CLR   (R0)+          ;CLEAR IT
1693 005764 162704 000002      SUB   #2,R4            ;2 BYTES LESS
1694 005770 001374          BNE   1$              ;MORE
1695 005772          POP   R4              ;RESTORE
                                MOV   (SP)+,R4

1696 005774          POP   R0              ;
                                MOV   (SP)+,R0

1697 005776 000207          RETURN          ;EXIT
1698 006000 000000 CLRPTR: .WORD
    
```

```

1701
1702      :++
1703      : SETUP - CALLED WITHIN EACH TEST TO INSERT BEGINNING ADDRESS OF THE
1704      : TEST INTO ALL UNITS TEST PC'S.
1705      : INPUTS:  TSTTOP LOADED WITH TEST ALGORITHMS STARTING ADDR.
1706      :          BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
1707      :          LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
1708      : OUTPUTS:  TSTPC(R5) FOR ALL UNITS
1709      :          DONE - CLEARED
1710      :--
1711
1712 006002 005037 003320  SETUP:: CLR      DONE      ;NOT DONE YET
1713 006006 012737 003346 003322  MOV      #BLKTBL, IDPTR ;TABLE TOP ADDR
1714 006014 017705 175302      1$:  MOV      @IDPTR, R5  ;DEVICE'S DATA BLOCK
1715 006020 013765 003324 000020  MCV     TSTTOP, TSTPC(R5);INSERT PC FOR TOP OF TEST
1716 006026 023727 003322 003364  CMP     IDPTR, #LSTDEV  ;ALL UNITS SET?
1717 006034 103004      BHIS    2$      ;YES
1718 006036 062737 000002 003322  ADD     #2, IDPTR      ;NO, GET NEXT POINTER
1719 006044 000763      BR      1$      ;SET HIM UP
1720 006046 000207      2$:  RETURN    ;DONE
    
```

```

1723          :++
1724          : RUN - IMPLEMENTS THE CALLS TO SEND PACKETS, RECEIVE PACKETS, THEN
1725          : CHECK ANSWERS DURING TEST RUN TIME.
1726          : INPUTS:  DONE
1727          : OUTPUTS: NONE
1728          :--
1729
1730 006050 004737 006100   RUN::  CALL  NXTST           ;MAKE AND SEND NEXT PACK TO ALL
1731                                     ;UNABORTED UNITS
1732 006054 005737 003320   TST    DONE           ;COMPLETE?
1733 006060 001006          BNE    2$             ;YES
1734 006062 004737 007150   CALL  GETANS         ;NO,GET ALL RESPONSES
1735
1736 006066          BREAK           ;SUPERVISOR CHECK
1737 006066 104422          TRAP    CSBRK
1738 006070 004737 010442   CALL  CHKANS        ;CHECK ALL RESPONSES
1739 006074 000765          BR    RUN           ;CONTINUE TILL DONE
1740 006076 000207   2$:  RETURN

```

```

1743      .SBTTL  NXTST / THE SCHEDULER
1744
1745      :++
1746      : NXTST - DISPATCH EXECUTION USING EACH UN-ABORTED UNIT'S TEST PROGRAM
1747      : COUNTER, (TSTPC(R5)). (THE POINTER TO THE TEST CODE THAT COMPRISES
1748      : MAKING A PACKET AND SENDING IT. CHECKS FIRST FOR ANY UN-ABORTED UNIT
1749      : THAT IS RETRYING EITHER A DATA ERROR OR A 'INDECIPHERABLE FLAG BYTE'
1750      : ERROR, IN ORDER TO SERVICE ONLY THAT UNIT THIS PASS.  INITS
1751      : NON-RETRYING UNITS IF NECESSARY.  IF NO RETRIES,DISPATCH ALL
1752      : UNITS IN ROUND ROBIN FASHION.
1753
1754      : INPUTS:      (IMPLIED) DATA BLOCKS.
1755      : BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
1756      : LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
1757
1758      : OUTPUTS:    ERRSF IF ALL UNITS ARE ABORTED.(TC NOTIFY APT)
1759      : SYSTAT IS UPDATED
1760
1761      :--
1762 006100 000240      NXTST:: NOP
1763 006102 012737      03346 003310  MOV      #BLKTBL,DEVPTR      ;UNIT 0 TO START
1764 006110 017705      75174      1$:  MOV      @DEVPTR,R5      ;GET DATA BLOCK.
1765 006114 005715      TST      @R5              ;ABORTED?
1766 006116 100504      BMI      2$              ; YES... CHECK NEXT UNIT
1767 006120 032715      000010  3$:  BIT      #BIT3,@R5      ;NO-RETRY 'BAD FLAG'?
1768 006124 001040      BNE      5$              ;YES...(SEND BREAK;THEN CMD PACK)
1769 006126 032715      020000  BIT      #BIT13,@R5     ;NO-RETRYING STILL (NO END PACK YET)?
1770 006132 001426      BEQ      7$              ;NO...
1771 006134 032715      000400  BIT      #BIT8,@R5     ;RETRYING A WRITE?
1772 006140 001453      BEQ      4$              ;NO...
1773 006142      SWAPIN      ;YES-GET DEVICE REGESTERS
1774      006142 016500      000006  MOV      6.(R5),R0
1775      006146 016501      000010  MOV      8.(R5),R1
1776      006152 016502      000012  MOV      10.(R5),R2
1777      006156 016503      000014  MOV      12.(R5),R3
1778      006162 016504      000016  MOV      14.(R5),R4
1774 006166 020265      000206  CMP      R2,SAVCNT(R5)  ;CURRENT COUNT = SAVED COUNT? (WHERE WE STARTED)
1775 006172 001036      BNE      4$              ;NO...(CONTINUE SENDING DATA PACKS)
1776 006174 042737      000004  003304  BIC      #BIT2,SYSTAT   ;YES-CLEAR RETRY FLAGS
1777 006202 042715      020000  BIC      #BIT13,@R5
1778 006206 000450      BR      2$              ;CHECK NEXT UNIT.
1779 006210 032715      002000  7$:  BIT      #BIT10,@R5    ;NO-RETRY DATA ERROR?
1780 006214 001445      BEQ      2$              ;NO...ON TO NEXT UNIT
1781 006216 052737      000002  003304  BIS      #BIT1,SYSTAT   ;SET RETRY STATUS TO 'DATA ERROR' TYPE
1782 006224 000424      BR      6$              ;YES...
1783
1784      006226      5$:  SWAPIN      ;GET DEVICE REGISTERS
1785      006226 016500      000006  MOV      6.(R5),R0
1786      006232 016501      000010  MOV      8.(R5),R1
1787      006236 016502      000012  MOV      10.(R5),R2
1788      006242 016503      000014  MOV      12.(R5),R3
1789      006246 016504      000016  MOV      14.(R5),R4
1785 006252 010265      000206  MOV      R2,SAVCNT(R5)  ;SAVE THE BYTE COUNT (FOR WRITE OPERATION)
1786
1787 006256 004737      013654  CALL     DOBRK          ;TO MARK HOW MANY DATA PACKS TO SEND
                          ;SEND INIT
  
```

```

1788 006262 032715 100000          BIT      #BIT15,@R5      :ABORTED?
1789 006266 001020          BNE      2$              :YES...
1790 006270 052737 000004 003304 4$:  BIS      #BIT2,SYSTAT   :NOT ABORTED-SET RETRY STATUS
1791 006276          SWAPIN          :GET DEVICE REGISTERS
      0C6276 016500 000006          MOV      6.(R5),R0
      006302 016501 000010          MOV      8.(R5),R1
      006306 016502 000012          MOV     10.(R5),R2
      006312 016503 000014          MOV     12.(R5),R3
      006316 016504 000016          MOV     14.(R5),R4

1792 006322 004775 000020          JSR      PC,@TSTPC(R5)  :DO TEST FOR
1793 006326 000477          BR      NXTRET         :THIS UNIT ONLY-EXIT
1794 006330 023727 003310 003364 2$:  CMP      DEVPTR,#LSTDEV :TRY NEXT UNIT?
1795 006336 103004          BHIS    NXTST2         :NO
1796 006340 062737 000002 003310  ADD      #2.,DEVPTR     :YES,->NEXT
1797 006346 000660          BR      1$              :GET BLOCK
1798
1799 006350 005037 006530 003310  NXTST2: CLR      ABONM    :HERE=NO RETRIES TO DO, NO UNIT ABORTED YET
1800 006354 012737 003346          MOV      #BLKTBL,DEVPTR :-->UNIT 0 STORAGE BLOCK
1801 006362 017705 174722  PERDEV: MOV      @DEVPTR,R5 :R5-->NEXT DEVICE STORAGE BLOCK
1802
1803 006366 005715          3$:  TST      @R5          :ABORTED?
1804 006370 100426          BMI      4$              :YES
1805 006372 032715 040000          BIT      #BIT14,@R5    :SEND BREAK?
1806 006376 001407          BEQ      6$              :NO
1807 006400 004737 013654          CALL    DOBRK          :YES
1808 006404 032715 040000          BIT      #BIT14,@R5    :SUCCESSFUL INIT?
1809 006410 001016          BNE      4$              :NO ON TO NEXT UNIT
1810 006412 005715          TST      @R5          :ABORTED?
1811 006414 100414          BMI      4$              :YES-ON TO NEXT UNIT
1812 006416          6$:  SWAPIN          :NO,GET DEVICE REGISTERS R0-R4 CONTAINING TEST PARAMETERS
      006416 016500 000006          MCV      6.(R5),R0
      006422 016501 000010          MOV      8.(R5),R1
      006426 016502 000012          MOV     10.(R5),R2
      006432 016503 000014          MOV     12.(R5),R3
      006436 016504 000016          MOV     14.(R5),R4

1813 006442 004775 000020          JSR      PC,@TSTPC(R5)  :INITIATE 1 PACKET TRANSMISSION AND RETURN
1814 006446 005715          4$:  TST      @R5          :ABORTED?
1815 006450 100002          BPL      8$              :NO-ON TO NEXT UNIT
1816 006452 005237 006530          INC      ABONM          :YES...ONE MORE TALLIED
1817 006456 023727 003310 003364 8$:  CMP      DEVPTR,#LSTDEV :ALL TU'S TRIED?
1818 006464 103004          BHIS    5$              :YES
1819 006466 062737 000002 003310  ADD      #2.,DEVPTR     :NO THE ADDRESS+2=NEXT ADDRESS
1820 006474 000732          BR      PERDEV         :DO NEXT UNIT
1821 006476 022737 000010 006530 5$:  CMP      #8.,ABONM     :ALL ABORTED?
1822 006504 001010          BNE      NXTRET        :NO
1823 006506          ERRSF  100.,NOMOR     :YES!
      006506 104454          TRAP    C$ERSF
      006510 000144          .WORD  100
      006512 006532          .WORD  NOMOR
      006514 000000          .WORD  0
1824 006516          11$: BREAK          :SUPERVISOR BREAK
      006516 104422          TRAP    C$BRK
1825 005520 005237 003336          INC      ALLGON         :SET DON'T-PRINT STATISTICS FLAG
1826 006524          DOCLN          :EXIT
      006524 104444          TRAP    C$DCLN
  
```

1827	006526	000207				NXTRET: RETURN	
1828							
1829	006530	000000				ABONM: .WORD	:THE NUMBER OF ABORTED UNITS
1830	006532	101	114	114		NOMOR: .ASCIZ	/ALL UNITS ABORTED!/ .EVEN
1831							

G
C

1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867 006556 000240
1868 006560 012665 000020
1869 006564
006564 010065 000006
006570 010165 000010
006574 010265 000012
006600 010365 000014
006604 010465 000016

1870
1871
1872 006610 022737 000002 003340
1873 006616 001007
1874 006620 022765 000000 000210
1875 006626 001523
1876 006630 012700 026174
1877 006634 000404
1878 006636 012700 026173
1879 006642 005265 000070
1880 006646 004737 007100
1881 006652 005715
1882 006654 100510
1883 006656 005365 000070

.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 (XSPKMM), 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.
 : XSPKMM(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 XSPKMM IS REPLICATED.
 : PACKET POINTER (PKPTR(R5)) POINTS TO TOP OF UNITS RECEIVE BUFFER
 : AREA (RCVBUF(R5)) FOR CURRENT UNIT.

RSVP:: NOP :FINISH TEST
 MOV (SP)+,TSTPC(R5) :SAVE WHERE YOU WERE IN TEST BODY AND
 SWAPOW :SAVE TEST REGISTERS
 MOV R0,6.(R5)
 MOV R1,8.(R5)
 MOV R2,10.(R5)
 MOV R3,12.(R5)
 MOV R4,14.(R5)

NOXOFF: MOV #TRBUF,R0 :FOR NORMAL PACKET SEND
 BR SND :SEND XOFF+PACKET
 XFNSND: MOV #TRBUF-1,R0 :POINT TO XOFF
 INC SNDCNT(R5) :ONE MORE TO SEND, TOO.
 SND: CALL SNDBYT :SEND BYTE
 TST @R5 :R5--> TO STATUS BLK
 BMI 6\$:ABORTED? YES...QUIT
 DEC SNDCNT(R5) :NO, SEND MORE

:CORRECT FOR RETURN TO SCHEDULER
 :***** IS THIS TEST 8
 :***** NO
 :***** IF SO, IS THIS UNIT MODIFIED
 :***** YES

```

1884 006662 001371          BNE      SND          ;IF MORE TO SEND
1885 006664 012700 026174    MOV     #TRBUF,R0     ;-->BUFFER
1886 006670 016537 000064 003330  MOV     REC(R5),BLKER ;PREPARE FOR RECEIVE
1887 006676 156565 000032 000033  BISB   XSPKNN(R5),XSPKNN+1(R5) ;REPLICATE LO. BYTE TO HI FOR GTPAKS, CHKANS
1888 006704 005065 000076          CLR     SUCCS(R5)     ;NO SUCCESS YET
1889 006710 042715 001000          BIC     #BIT9,@R5     ;NO DATA CHK ERROR YET
1890 006714 016565 000102 000104  MOV     RCVBUF(R5),PKPTR(R5) ;TOP OF RCV BUFFER GOES THE 1ST PACKET
1891 006722 012704 00G034          MOV     #XSFLG,R4     ;FORM
1892 006726 060504          ADD     R5,R4         ;ADDRESS
1893 006730 010465 000106          MOV     R4,XSPTR(R5)  ;OF 1ST XSFLG
1894
1895 006734 042715 000020          BIC     #BIT4,@R5     ;PRESET AS DATA PAK
1896 006740 121027 000002          CMPB   @R0,#RSCMND   ;WAS IT COMMAND PAK?
1897 006744 001054          BNE     6$           ;NO...
1898 006746 116065 000002 000100  MOVB   2(R0),CMDSNT(R5) ;YES-SAVE COMMAND
1899 006754 052715 000020          BIS    #BIT4,@R5     ;ITS CMND PAK
1900
1901 006760 032715 002000          BIT    #BIT10,@R5    ;RETRYING?
1902 006764 001044          BNE     6$           ;YES-DON'T UPDATE ANY STATS OR CONDITION
1903 006766 126027 000002 000002  CMPB   2(R0),#RSSRD  ;NO,A READ?
1904 006774 001012          BNE     4$           ;NO
1905 006776 042715 000400          BIC    #BIT8,@R5     ;(FOR HARD/SOFT LOGGING) RD/WR FLAG=0
1906 007002 004737 013504          CALL   WHCHDR        ;GET DRIVE
1907 007006 103403          BCS    8$           ;
1908 007010 005265 000114          INC    RDNO(R5)      ;DRIVE 0
1909 007014 000402          BR     4$           ;
1910 007016 005265 000116          8$:    INC    RDN1(R5)  ;DRIVE 1
1911
1912 007022 126027 000002 000003  4$:    CMPB   2(R0),#RSSWR  ;A WRITE?
1913 007030 001022          BNE     6$           ;NO
1914 007032 052715 000400          BIS    #BIT8,@R5     ;YES, RD/WR FLAG=1
1915 007036 105760 000003          TSTB   3(R0)         ;VERIFY TOO?
1916 007042 001403          BEQ    21$          ;NO
1917 007044 052715 000040          BIS    #BIT5,@R5     ;YES-SET VERIFY FLAG
1918 007050 000402          BR     22$          ;
1919 007052 042715 000040          21$:   BIC    #BIT5,@R5     ;(NO)-RESET VERIFY FLAG
1920 007056 004737 013504          22$:   CALL   WHCHDR        ;GET DRIVE NO
1921 007062 103403          BCS    5$           ;CARRY=DR1
1922 007064 005265 000110          INC    WRTNO(R5)     ;# BLKS WRITTEN DRO
1923 007070 000402          BR     6$           ;EXIT
1924
1925 007072 005265 000112          5$:    INC    WRTN1(R5)  ;# BLKS WRITTEN DRV1
1926 007076          6$:
1927 007076 000207          ENDRSP: RETURN      ;RETURN
  
```

```

1930      .SBTTL  SNDBYT / OUTPUT A BYTE TO UNIT
1931
1932      :++
1933      : SNDBYT - TEST 'READY' ON INTERFACE.  IF 'READY', SEND BYTE AND EXIT.
1934      :           IF TIMED OUT, LOG ERROR.
1935      : INPUTS - RO = POINTER TO BUFFER
1936      :           - IMPLIED UNIT DATA BLOCK
1937      :           - CSNRDY - TIMEOUT CONSTANT
1938      : OUTPUTS - RO IS INCREMENTED.
1939      : ERROR - NOT-READY-TO-SEND TIME OUT
1940      :--
1941
1942 007100 SNDBYT:: PUSH  R1 .                ;ENTER RO-->BYTE
      007100 010146                               MOV    R1,-(SP)

1943 007102 013701 003342      4$:  MOV    CSNRDY,R1          ;GET TIMEOUT CONSTANT FOR NOT READY ERROR
1944 007106 105775 000026      1$:  TSTB  @XMSR(R5)         ;READY TO SEND?
1945 007112 100412              BMI   2$                    ;YES
1946 007114 010046              PUSH  R0                     ;NO, SAVE R0
                                      MOV    R0,-(SP)

1947 007116              BREAK          ;MONITOR BREAK
      007116 104422              TRAP   C$BRK
1948 007120              POP    R0          ;RESTORE
      007120 012600              MOV    (SP)+,R0

1949
1950 007122 005301              DEC   R1                    ;ABORTED?
1951 007124 001370              BNE  1$                    ;NO
1952 007126 012704 000056      MOV   #TOSNDB,R4          ;YES,SET CODE FOR TIMEOUT ERROR
1953 007132 004737 012500      CALL LOG                 ;LOG IT
1954 007136 000402              BR   3$                    ;QUIT
1955 007140 112075 000030      2$:  MOVB  (R0)+,@XMDB(R5) ;SEND IT
1956 007144 012601              3$:  POP   R1                    ;RESTORE
                                      MOV    (SP)+,R1

1957 007146 000207              RETURN          ;DONE
  
```

```

1960 .SBTTL GETANS / GETS RESPONSES ROUND ROBIN USING "XON"
1961
1962
1963 :++
1964 : GETANS - IF A UNIT IS RETRYING CLEAR HIS RECEIVE BUFFER (CLRBUF) AND GET
1965 : HIS RESPONSE (GTPKS1), ELSE, CLEAR ALL BUFFERS (CLRALL) AND
1966 : GET ALL RESPONSES (GTPKS8).
1967 : INPUTS: SYSTAT - SYSTEM STATUS WORD.
1968 : OUTPUTS: SERVST = -1 IF NO RETRIES.
1969 :--
1970
1971 007150 000240 GETANS:: NOP ;1 UNIT IF RETRY; ELSE ALL
1972 007152 032737 000006 003304 BIT #BIT1!BIT2,SYSTAT ;RETRY?
1973 007160 001010 BNE 1$ ;YES
1974 007162 012737 177777 010206 MOV #-1,SERVST ;PRESET NO UNITS SERVICED
1975 007170 004737 005706 CALL CLRALL ;CLEAR ALL INPUT BUFFERS
1976 007174 004737 007426 CALL GTPKS8 ;GET ALL REPLYs
1977 007200 000404 BR 2$ ;EXIT
1978 007202 004737 005746 1$: CALL CLRBUF ;RETRY-CLEAR 1 UNIT ONLY
1979 ;R5->UNIT BY NXTST
1980 007206 004737 007216 CALL GTPKS1 ;GET 1 REPLY
1981 007212 000207 2$: RETURN ;DONE
1982
1983 007214 000000 GETPTR: .WORD
  
```

G
C

1986
 1987
 1988
 1989
 1990
 1991
 1992
 1993
 1994
 1995
 1996
 1997
 1998
 1999
 2000 007216 000240
 2001 007220 012703 000034
 2002 007224 060503
 2003 007226 010301
 2004 007230 062701 000002
 2005 007234 012700 007424
 2006 007240 004737 007100
 2007
 2008 007244 016500 000102
 2009 007250 116502 000033
 2010 007254 032702 177400
 2011 007260 011137 003314
 2012 007264 011337 003312
 2013 007270 004737 010212
 2014 007274 032715 100000
 2015 007300 001050
 2016 007302 005300
 2017 007304 111037 003305
 2018 007310 121037 003312
 2019 007314 001420
 2020 007316 121027 000002
 2021 007322 001006
 2022 007324 012737 000016 003314
 2023 007332 012702 000001
 2024 007336 000407
 2025 007340 121027 000001 14\$:
 2026 007344 001026
 2027 007346 012737 000204 003314
 2028 007354 065202
 2029
 2030 007356 005200 2\$:
 2031 007360 005337 003314 5\$:
 2032 007364 001411
 2033 007366 004737 010212
 2034 007372 005765 000074
 2035 007376 001011
 2036 007400 032715 100000
 2037 007404 001006
 2038 007406 000764
 2039
 2040 007410 005302 3\$:
 2041 007412 001403
 2042

.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
:--
  
```

```

GTPKS1:: NOP ;R5->THE UNIT
          MOV  #XSFLG,R3 ;THE OFFSET VALUE OF FLAG
          ADD  R5,R3 ;FORM THE ABSOLUTE ADDRESS
          MOV  R3,R1 ;R3-->ADDR. OF EXPECTED FLAG
          ADD  #2,R1 ;R1-->ADDR. OF EXPECTED COUNT
          MOV  #EXON,R0 ;R0=ADDRESS
          CALL SNDBYT ;XON THE DEVICE
          ;*** TIME CRITICAL
          MOV  RCVBUF(R5),R0 ;***--> TO THE BUFFER
          MOVB XSPKMM+1(R5),R2 ;***GET THE # OF PACKETS TO RECEIVE
          BIT  #177400,R2 ;***SIGN UN-EXTEND
1$: MOV  @R1,RCBCNT ;***HOW MANY BYTES IT SHOULD BE
     MOV  @R3,RCFLG ;***WHAT THE FIRST BYTE SHOULD BE
     CALL GTBYTE ;***GET THE ALL IMPORTANT FLAG
     BIT  #BIT15,@R5 ;TIMEOUT?
     BNE  4$ ;YES
     DEC  R0 ;-> BYTE RECIEVED
     MOVB @R0,SYSTAT+1 ;SAVE IT AS FLAG BYTE
     CMPB @R0,RCFLG ;1ST BYTE WHAT WAS EXPECTED?
     BEQ  2$ ;YES
     CMPB @R0,#RSEND ;NO, WAS IT END PAK?
     BNE  14$ ;NO
     MOV  #RSNDSZ,RCBCNT ;YES, USE END SIZE FOR COUNT
     MOV  #1,R2 ;AND ASSUME IT'S LAST PACKET!
     BR   2$ ;CONTINUE RECEIVE
14$: CMPB @R0,#RSDATA ;WAS IT DATA?
     BNE  4$ ;NO,CHKANS MAY FIND INIT...
     MOV  #RSDASZ,RCBCNT ;YES, SET FOR DATA PAK SIZE
     INC  R2 ;ONE MORE PACK THAN EXPECTED (END PAK)
2$: INC  R0 ;RESTORE TO -> MEXT BYTE
5$: DEC  RCBCNT ;THAT'S ONE LESS BYTE TO GO
     BEQ  3$ ;DONE
     CALL GTBYTE ;GET REST OF PACKET
     TST  DLV(R5) ;ERROR
     BNE  4$ ;YES-ALL OVER
     BIT  #BIT15,@R5 ;OR IF ABORTED
     BNE  4$ ;THEN QUIT
     BR   5$ ;CONTINUE RECEIVE
3$: DEC  R2 ;ONE LESS PACKET TO GO
     BEQ  4$ ;MORE PACKETS IN TRANSACTION?
     ;YES
  
```

2043	007414	022121	CMP	(R1)+,(R1)+	:POINT TO NEW EXPECTED COUNT
2044	007416	022323	CMP	(R3)+,(R3)+	:AND FLAG,
2045	007420	000717	BR	1\$:AND RECEIVE,
2046	007422	000207	4\$: RETURN		:RETURN
2047					
2048	007424	020	EXON:	.BYTE	RSXON
2049	007425	023	EXOFF:	.BYTE	RSXOFF

2052
 2053
 2054
 2055
 2056
 2057
 2058
 2059
 2060
 2061
 2062
 2063
 2064
 2065
 2066
 2067
 2068
 2069
 2070
 2071 007426 000240
 2072 007430 022737 000002 003340
 2073 007436 001006
 2074 007440 022765 000000 000210
 2075 007446 001002
 2076 007450 000137 010114
 2077 007454 012737 003346 010210
 2078 007462 017705 000522
 2079 007466 032715 100000
 2080 007472 001403
 2081 007474 004737 010122
 2082 007500 000564
 2083 007502 105765 000033
 2084 007506 001003
 2085 007510 004737 010122
 2086 007514 000556
 2087 007516 105365 000033
 2088 007522 017537 000106 003312
 2089 007530 062765 000002 000106
 2090 007536 017537 000106 003314
 2091 007544 022737 000002 003340
 2092 007552 001404
 2093 007554 012700 007424
 2094
 2095 007560 004737 007100
 2096 007564 016500 000104
 2097 007570 004737 010212
 2098 007574 032715 100000
 2099 007600 001403
 2100 007602 105065 000033
 2101 007606 000521
 2102 007610 005300
 2103 007612 111037 003305
 2104 007616 121037 003312
 2105 007622 001436
 2106 007624 105065 000033
 2107 007630 121027 000002
 2108 007634 001004

```
.SBTTL GTPKS8 / GET RESPONSES (NO RETRIES)
:++
: GTPKS8 - IF IN TEST 8 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, DECREMENDING 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 8, SEND 'XOFF' TO ENHANCE THROUGHPUT AND GO ON
: TO NEXT UNIT (IF ANY). IF IN TEST 8, 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
:--
GTPKS8:: NOP ;GET ALL UNITS RESPONSES XOFF IF DATA PAK (THROUGHPUT)
          CMP #2,TEST8 ;***** IS THIS TEST 8
          BNE 1$ ;***** NO
          CMP #0,MRSP(R5) ;***** IF SO, IS THIS UNIT MODIFIED
          BNE 1$ ;***** YES, CONTINUE NORMALLY
          JMP ENDGP8 ;***** ELSE, SKIP ROUTINE
1$:      MOV #BLKTBL,GPTR ;->1ST
GTAGIN: MOV @GTPTR,R5 ;GET DATA BLOCK
          BIT #BIT15,@R5 ;ABORTED?
          BEQ 2$ ;NO
          CALL SETSRV ;YES-SET 'SERVICED' AND
          GTDOWN ;ON TO NEXT UNIT
2$:      TSTB XSPKNM+1(R5) ;NO, ANY PACKETS LEFT?
          BNE 3$ ;YES
          CALL SETSRV ;NO-HE'S DONE
          BR GTDOWN ;SO ON TO NEXT UNIT
3$:      DECB XSPKNM+1(R5) ;NOW ITS ONE LESS PACKET
          MOV @XSPTR(R5),RCFLG ;GET EXPECTED FLAG
          ADD #2,XSPTR(R5) ;--> COUNT
          MOV @XSPTR(R5),RCBCNT ;AND EXPECTED COUNT
          CMP #2,TEST8 ;***** IF TEST 8
          BEQ 1$ ;***** DO NOT SEND XON
          MOV #EXON,R0 ;-> XON
          ;***TIME CRITICAL
          CALL SNDBYT ;***SEND IT
1$:      MOV PKPTR(R5),R0 ;***->WHERE 1ST BYTE GOES
          CALL GTBYTE ;***GET IT
          BIT #BIT15,@R5 ;ABORTED?
          BEQ 4$ ;NO-CONTINUE
          CLRB XSPKNM+1(R5) ;YES-NO MORE PACKETS EXPECTED
          BR GTDOWN ;ON TO NEXT
4$:      DEC R0 ;-->BYTE JUST RECEIVED
          MOVB @R0,SYSTAT+1 ;SAVE IT
          CMPB @R0,RCFLG ;IS IT WHAT EXPECTED?
          BEQ GTOK ;YES
          UNXPCT: CLRB XSPKNM+1(R5) ;NO, MUST BE LAST REPLY
          CMPB @R0,#RSEND ;MAYBE AN END PAK?
          BNE 4$ ;NO
```

```

2109 007636 012737 000016 003314      MOV      #RSNDSZ,RCBCNT ;YES, USE PROPER COUNT
2110 007644 000406                    BR       GTUM          ;AND GET IT
2111 007646 121027 000001              4$:     CMPB     @R0,#RSDATA ;IS IT DATA?
2112 007652 001077                    BNE     GTDOWN        ;NO, ALL OVER, CHKANS WILL INIT UNIT
2113 007654 012737 000222 003314      MOV      #RSDNSZ,RCBCNT ;YES, USE COUNT OF DATA + END PAK SURE TO FOLLOW
2114 007662 005200                    GTUM:   INC      RO      ;WHERE TO STUFF THE REST
2115 007664 005337 003314              5$:     DEC      RCBCNT     ;ONE DOWN
2116 007670 001470                    BEQ     GTDOWN        ;NONE TO GO
2117 007672 004737 010212                    CALL    GTBYTE        ;MORE TO GO
2118 007676 032715 100000                    BIT     #BIT15,@R5    ;TIMEOUT?
2119 007702 001063                    BNE     GTDOWN        ;YES
2120 007704 005765 000074                    TST    DLV(R5)       ;BUT DLV ERROR?
2121 007710 001765                    BEQ     5$           ;NO
2122 007712 105065 000033                    CLRB   XSPKMN+1(R5)  ;YES-LAST TIME
2123 007716 000455                    BR      GTDOWN        ;ON TO NEXT
2124
2125 007720 005200                    GTOK:   INC      RO      ;NEXT PLACE IN BUFFER
2126 007722 022737 000002 003340      1$:     CMP      #2,TEST8   ;***** IF NOT TEST 8
2127 007730 001020                    BNE     7$           ;***** DO NOT SEND 'CONT'
2128 007732 010046                    PUSH   RO            ;*****SEND 'CONT' FOR MRSP
                               MOV      RO,-(SP)

2129 007734 012700 010116                    MOV      #MODRSP,RO  ;
2130 007740 004737 007100                    CALL    SNDBYT      ;
2131 007744 000240                    NOP
2132 007746 012737 000001 010120      MOV      #1,MRSPLY   ;***** ANOTHER DELAY
2133 007754 005000                    2$:     CLR      RO      ;*****
2134 007756 005300                    3$:     DEC      RO      ;***** TO GET AN ERROR
2135 007760 001376                    BNE     3$           ;***** IF MRSP DOESN'T WORK
2136 007762 005337 010120                    DEC     MRSPLY      ;*****
2137 007766 001372                    BNE     2$           ;
2138 007770 012600                    POP     RO          ;
                               MOV      (SP)+,RO

2139 007772 005337 003314              7$:     DEC      RCBCNT     ;MORE BYTES?
2140 007776 001413                    BEQ     4$           ;NO-ALL DONE
2141 010000 004737 010212                    CALL    GTBYTE        ;YES-GET IT
2142 010004 032715 100000                    BIT     #BIT15,@R5    ;TIMEOUT?
2143 010010 001020                    BNE     GTDOWN        ;YES
2144 010012 005765 000074                    TST    DLV(R5)       ;ERROR?
2145 010016 001741                    BEQ     1$           ;NO
2146 010020 105065 000033                    CLRB   XSPKMN+1(R5)  ;LAST TIME
2147 010024 000412                    BR      GTDOWN        ;EXIT
2148 010026 122775 000001 000104      4$:     CMPB     #RSDATA,@PKPTR(R5) ;WAS DATA?
2149 010034 001006                    BNE     GTDOWN        ;NO, ALL DONE
2150 010036 010065 000104                    MOV     RO,PKPTR(R5) ;START OF NEXT PACK NEXT TIME
2151 010042 012700 007425                    MOV     #EXOFF,RO    ;XOFF AND SEND TO
2152 010046 004737 007100                    CALL    SNDBYT      ;ENHANCE THROUGHPUT
2153 010052 062765 000002 000106      GTDOWN: ADD     #2,XSPTR(R5) ;NEXT XSFLG FOR NEXT TRY
2154 010060 023727 010210 003364      CMP     GTPTR,#LSTDEV ;DONE ONE CYCLE ALL UNITS?
2155 010066 103005                    BHIS   1$           ;YES
2156 010070 062737 000002 010210      ADD     #2,GTPTR     ;NEXT UNIT
2157 010076 000137 007462                    JMP     GTAGIN       ;CONTINUE RECEIVE
2158 010102 105737 010206              1$:     TSTB    SERVST   ;DONE SERVICING ALL PAKS
2159
2160 010106 001402                    BEQ     ENDGP8      ;FROM ALL UNITS?
                               ;YES

```

2161	010110	000137	007426		JMP	GTPKS8		;NO, KEEP TRYING
2162	010114	000207		ENDGP8:	RETURN			;RETURN
2163								
2164	010116	020		MODRSP:	.BYTE	RSCONT		
2165					.EVEN			
2166	010120	000000		MRSPLY:	.WORD			

G
L

```

2169          .SBTTL  SETSRV / SET UNIT SERVICED
2170
2171          :++
2172          : SETSRV - RESET THE BIT IN 'SERVST' CORRESPONDING TO THE UNIT NUMBER.
2173          : INPUTS - SERVST - 'SERVICED' WORD
2174          :           - @R5 = UNIT # (BITS 0, 1, 2)
2175          : OUTPUTS - SERVST MODIFIED
2176          :--
2177
2178 010122     SETSRV: PUSH      R5          ;SET UNIT SERVICED
          010122     010546          MOV      R5,-(SP)

2179 010124     PUSH      R0          MOV      R0,-(SP)
          010124     010046

2180 010126     011505          MOV      @R5,R5          ;GET STAT WD
2181 010130     042705     177770    BIC      #177770,R5     ;MASK UNIT #
2182 010134     012700     010166    MOV      #SRVTBL,R0     ;->TOP OF BIT TABLE
2183 010140     005705          1$: TST      R5          ;RIGHT ONE?
2184 010142     001404          BEQ      2$          ;YES
2185 010144     062700     000002    ADD      #2,R0         ;NO, ->NEXT
2186 010150     005305          DEC      R5          ;1 LESS
2187 010152     000772          BR       1$          ;CONTINUE
2188 010154     041037     010206    2$: BIC      @R0,SERVST ;MOW IT DOWN
2189 010160     010160     012600    POP      R0          MOV      (SP)+,R0

2190 010162     010162     012605    POP      R5          MOV      (SP)+,R5

2191 010164     000207          RETURN          ;RETURN
2192
2193 010166     000001          SRVTBL: .WORD  BIT0     ;BIT POSITION LOOKUP TABLE
2194 010170     000002          .WORD  BIT1
2195 010172     000004          .WORD  BIT2
2196 010174     000010          .WORD  BIT3
2197 010176     000020          .WORD  BIT4
2198 010200     000040          .WORD  BIT5
2199 010202     000100          .WORD  BIT6
2200 010204     000200          .WORD  BIT7
2201
2202 010206     000000          SERVST: .WORD
2203 010210     000000          GTPTR:  .WORD
    
```

2206
 2207
 2208
 2209
 2210
 2211
 2212
 2213
 2214
 2215
 2216
 2217
 2218
 2219
 2220
 2221
 2222
 2223
 2224
 2225
 2226
 2227
 2228
 2229 010212 005037 010436
 2230 010216 013704 003344
 2231 010222 105775 000022
 2232 010226 100013
 2233 010230 017565 000024 000074
 2234 010236 116520 000074
 2235 010242 005765 000074
 2236 010246 100472
 2237 010250 005065 000074
 2238 010254 000467
 2239 010256 005337 010436
 2240 010262 001357
 2241
 2242
 2243
 2244 010264 010037 010440
 2245 010270 012700 007425
 2246 010274 004737 007100
 2247 010300 105775 000022
 2248 010304 100415
 2249 010306 005337 010436
 2250 010312 105737 010436
 2251 010316 001370
 2252 010320
 2253 010322 012700 007424
 2254 010326 004737 007100
 2255 010332 013700 010440
 2256 010336 000426
 2257 010340 013700 010440 000074
 2258 010344 017565 000024
 2259 010352 116520 000074
 2260 010356 005765 000074
 2261 010362 100403

.SBTTL GTBYTE / GET A BYTE FROM UNIT

```

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

```

: INPUTS - R0 POINTS TO INPUT BUFFER
          - IMPLIED UNITS DATA BLOCK
          - CSRCVB TIME OUT MULTIPLIER
    
```

```

: OUTPUTS - R0 IS INCREMENTED
           - DLV (R5) NON-ZERO ON INTERFACE ERROR.
    
```

```

: ERROR - TIME OUT ON RECEIVE
:--
    
```

```

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

;CODE TO SEE ^C DURING LONG SEEK OR REWIND
          MOV      R0,GBTMP2  ;HERE GBTMP=0
          MOV      #EXOFF,R0 ;R0 MUST BE PRESERVED!
          CALL     SNDBYT    ;QUIET THE DEVICE
          ;BY SENDING XOFF
6$:      TSTB     @RCSR(R5)   ;CHARACTER SLOP OVER?
          BMI      5$        ;YES
          DEC      GBTMP      ;NO-WAIT A WHILE
          TSTB     GBTMP      ;DONE WAITING?
          BNE     6$        ;NO
          BREAK    ;YES-NO SLOP OVER
          TRAP     CSBRK

          MOV      #EXON,R0   ;START DEVICE TALKING
          CALL     SNDBYT    ;AGAIN
          MOV      GBTMP2,R0 ;RESTORE R0
          BR       7$        ;END KLUGE
5$:      MOV      GBTMP2,R0  ;RESTORE R0
          MOV      @RCDDB(R5),DLV(R5) ;GET ERROR + BYTE
          MOVB     DLV(R5),(R0)+ ;COPY BYTE TO BUFFER
          TST      DLV(R5)    ;ERROR?
          BMI      17$       ;YES-EXIT
    
```

```
2262 010364 005065 000074          CLR    DLV(R5)          ;NO-CLEAR
2263 010370 000400                   BR     17$             ;EXIT
2264 010372 010037 010440          17$:  MOV    R0,GBTMP2   ;AGAIN SAVE R0
2265 010376 012700 007424          MOV    #EXON,R0       ;RESTORE TO TALKING STATE
2266 010402 004737 007100          CALL   SNDBYT         ;BY SENDING 'XON'
2267 010406 013700 010440          MOV    GBTMP2,R0      ;RESTORE R0
2268 010412 000410                   BR     4$              ;DONE
2269 010414 005037 010436          7$:   CLR    GBTMP
2270 010420 005304                   DEC    R4              ;TIMEOUT?
2271 010422 001277                   BNE   1$              ;NO
2272 010424 012704 000050          MOV    #TORCVB,R4    ;YES
2273 010430 004737 012500          CALL   LOG            ;LOG ERROR.
2274 010434 000207                   4$:   RETURN          ;RETURN
2275 010436 000000          GBTMP: .WORD 0
2276 010440 000000          GBTMP2: .WORD 0
```

2279
 2280
 2281
 2282
 2283
 2284
 2285
 2286
 2287
 2288
 2289
 2290
 2291
 2292
 2293
 2294
 2295
 2296
 2297
 2298
 2299
 2300
 2301
 2302
 2303
 2304
 2305
 2306
 2307
 2308
 2309
 2310
 2311
 2312
 2313
 2314
 2315
 2316

010442 000240
 010444 032737 000006 003304
 010452 001403
 010454 004737 010552
 010460 000432
 010462 012737 003346 010550
 010470 017705 000054
 010474 032715 100000
 010500 001012
 010502 022737 000002 003340
 010510 001004
 010512 022765 000000 000210
 010520 001402
 010522 004737 010552
 010526 023727 010550 003364
 010534 103004
 010536 062737 000002 010550
 010544 000751
 010546 000207
 010550 000000

.SBTTL CHKANS / CHECK DEVICE(S) RESPONSE

```

:++
:  CHKANS - AS IN 'GETANS', IF RETRYING DO ONLY 1 UNIT ELSE DO ALL NON-
:          ABORTED UNITS. NOTE, IF IN TEST 8 AND THE UNIT IS NOT
:          MODIFIED DO NOT CHECK UNIT.
:  INPUTS: IMPLIED SYSTAT BIT1 (RETRYING)
:          BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
:          LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
:  OUTPUTS: NONE PASSED.
:--
  
```

```

CHKANS:: NOP                                ;IF RETRY THEN CHECK ONE
;ELSE CHECK ALL
;RETRYING?
BIT #BIT1!BIT2,SYSTAT                       ;NO DO NORMAL
BEQ CHK8                                     ;YES DO SINGLE UNIT
CALL CHKPKS                                  ;R5 -> UNIT
BR CHKANR                                    ;ALL DONE

CHK8: MOV #BLKTBL,CHKPTR                    ;YOU KNOW ... TOP OF TABLE
2$: MOV @CHKPTR,R5                          ;GET UNIT'S BLOCK ADDRESS
BIT #BIT15,@R5                              ;ABORTED?
BNE 3$                                       ;YES
CMP #2,TEST8                                ;***** IS THIS TEST 8
BNE 1$                                       ;***** NO-CONTINUE NORMALLY
CMP #0,MRSP(R5)                             ;***** IF SO, IS THIS UNIT MODIFIED
BEQ 3$                                       ;***** NO SKIP NEXT INSTR
1$: CALL CHKPKS                              ;NO, DO THIS GUY
3$: CMP CHKPTR,#LSTDEV                      ;ALL DONE?
BHS CHKANR                                   ;YES
ADD #2,CHKPTR                                ;NO,-->NEXT DEVICE
BR 2$                                        ;DO DA

CHKANR: RETURN
CHKPTR: .WORD
  
```

2319
 2320
 2321
 2322
 2323
 2324
 2325
 2326
 2327
 2328
 2329
 2330
 2331
 2332
 2333
 2334
 2335
 2336
 2337
 2338
 2339
 2340
 2341
 2342
 2343
 2344 010552 000240
 2345 010554 042715 000010
 2346 010560 016500 000102
 2347 010564 116502 000032
 2348 010570 012703 000034
 2349 010574 060503
 2350 010576 010301
 2351 010600 062701 000002
 2352 010604 010065 000104
 2353 010610 111037 003305
 2354 010614 011137 003314
 2355 010620 011337 003312
 2356 010624 121013
 2357 010626 001057
 2358 010630 121027 000020
 2359 010634 001534
 2360
 2361 010636 013704 003314
 2362 010642 005744
 2363 010644 004737 013614
 2364 010650 103005
 2365 010652 012704 000022
 2366 010656 004737 012500
 2367 010662 000521
 2368 010664 122710 000002
 2369 010670 001005
 2370 010672 004737 011146
 2371 010676 012702 000001
 2372 010702 000511
 2373 010704 122710 000001
 2374 010710 001012
 2375 010712 022737 000001 003340

.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 8 A GET CHARACTERISTICS COMMAND PACKET WAS
SENT TO THE TUSB. IF THE RESPONSE WAS A DATA PACKET, WHICH IS
EXPECTED, THEN THE UNIT IS NOT MODIFIED, AND THE MRSP FLAG IS
CLEARED. IF THE RESPONSE IS AN END PACKET, WHICH WOULD BE
HANDLED BY THIS ROUTINE AS AN UNKNOWN, THEN THE UNIT IS MODIFIED,
AND THE MRSP FLAG IS SET.
INPUTS: (IMPLIED) UNITS DATA BLOCK
OUTPUTS: ERRORS - DLV ERROR
              - UNKNOWN FLAG BYTE ERROR
              - CHECKSUM ERROR
              - DATA COMPARE ERROR
R4 = ERROR NUMBER
SYSTAT UPPER BYTE = 1ST BYTE OF RESPONSE
:--
  
```

```

CHKPKS:: NOP ;CHECK WHAT WAS RECIEVED
          BIC #BIT3,@R5 ;CLEAR 'BAD FLAG' RETRY BIT
          MOV RCVBUF(R5),R0 ;GET BUFFER ADDR.
          MOV XSPKNT(R5),R2 ;AND # OF PACKETS EXPECTED
          MOV #XSFLG,R3 ;THE OFFSET VALUE
          ADD R5,R3 ;R3-->THIS UNIT XSFLG AGAIN
          MOV R3,R1 ;COPY TO R1
          ADD #2,R1 ;R1-->XSBCNT FOR 1ST PACKET
1$: MOV R0,PKPTR(R5) ;POINT TO PACKET
     MOV @R0,SYSTAT+1 ;SAVE RCV'D BYTE
     MOV @R1,RCBCNT ;GET COUNT
     MOV @R3,RCFLG ;AND FLAG
     CMPB @R0,@R3 ;1ST BYTE=EXPECTED?
     BNE 5$ ;UH OH...
     CMPB @R0,#RSCONT ;OK, IS IT 1 BYTE?
     BEQ 7$ ;YES...ONTO NEXT PACK
          ;NO, SO > 1 BYTE (NEVER EXPECT INIT!)
          ;EXPECTED, SO COUNT MUST BE RIGHT
          ;ADJUST FROM RECEIVE COUNT TO COUNT FOR CHECKSUM
          MOV RCBCNT,R4
          TST -(R4)
          CALL CKCKSM ;CHECK CHECKSUM
          BCC 2$ ;NO CARRY...NO INCORRECT
          MOV #BDCHK,R4 ;ERROR
          CALL LOG ;LOG IT
          BR 7$ ;ON TO NEXT PACK
2$: CMPB #RSEND,(R0) ;END PAK?
     BNE 3$ ;NO
     CALL CHKEND ;YES-CHECK
     MOV #1,R2 ;LAST PACKET
     BR 7$ ;AND FALL THROUGH
3$: CMPB #RSDATA,@R0 ;DATA PAK?
     BNE 4$ ;NO
     CMP #1,TEST8 ;***** IS THIS TEST 8
  
```

```

2376 010720 001003          BNE      11$          :***** NO-CONTINUE NORMALLY
2377 010722 005065 000210  CLR      MRSP(R5)   :***** CLR MRSP FLAG
2378 010726 000402          BR       12$          :***** SKIP INSTR
2379 010730 004737 014414  11$:  CALL    COMPAR   :YES-CHECK DATA
2380 010734 000474          12$:  BR       7$       :ALL DONE?
2381 010736 052715 020010  4$:   BIS     #BIT3!BIT13,@R5 :SET 'BAD FLAG' RETRY FLAGS
2382 010742 012704 000010  MOV     #OTL,R4     :OUT TO LUNCH
2383 010746 005765 000074  TST    DLV(R5)     :AH,BUT DLV ERROR?
2384 010752 001402          BEQ     20$          :NO
2385 010754 012704 000012  MOV     #OVRN,R4    :YES-USE CORRECT ERROR #
2386 010760 004737 012500  20$:  CALL    LOG       :TALLY
2387 010764 000467          BR      8$          :DONE
2388
2389                      ;HERE CHECKS UNEXPECTED RESPONSE
2390
2391 010766 122710 000004  5$:   CMPB   #RSINIT,@R0 :INIT?
2392 010772 001007          BNE     6$          :NO
2393 010774 052715 020010  BIS     #BIT3!BIT13,@R5 :YES-SET RETRY FLAGS
2394 011000 012704 000006  MOV     #RCINIT,R4   : WE GOT AN INIT
2395 011004 004737 012500  CALL    LOG          :TALLY IT
2396 011010 000455          BR      8$          :DONE
2397 011012 122710 000001  6$:   CMPB   #RSDATA,@R0 :DATA PAK?
2398 011016 001013          BNE     9$          :NO
2399 011020 012704 000204  MOV     #RSDASZ,R4   :YES, USE DATA SIZE
2400 011024 005744          TST    -(R4)        :ADJUST FOR CHKSUM
2401 011026 004737 013614  CALL    CKCKSM       :AND CHECK
2402 011032 103430          BCS    10$          :GOOF
2403 011034 004737 014414  CALL    COMPAR       :OK, HOW'S THE DATA?
2404
2405                      :EXPECTED END, GOT
2406 011040 062700 000204  ADD     #RSDASZ,R0   :DATA + END.
2407 011044 000657          BR      1$          :POINT TO END PACK
2408                      :CHECK IT, USE SAME XSFLG
2409 011046 122710 000002  9$:   CMPB   #RSEND,(R0) :END?
2410 011052 001331          BNE     4$          :NO-OUT TO LUNCH
2411 011054 012704 000016  MOV     #RSSNSZ,R4   :YES, TOTAL SIZE MINUS
2412 011060 005744          TST    -(R4)        :TWO (THE CHKSUM)
2413 011062 004737 013614  CALL    CKCKSM       :CHECK IT
2414 011066 103412          BCS    10$          :OOPS
2415 011070 022737 000001 003340  CMP     #1,TEST8     :***** IS THIS TEST 8
2416 011076 001003          BNE     13$         :***** NO-CONTINUE NORMALLY
2417 011100 012765 000001 000210  MOV     #1,MRSP(R5)  :***** IF SO, SET THE MRSP FLAG
2418 011106 004737 011146  13$:  CALL    CHKEND      :OK,NOW TEST SUC. CODE
2419
2420 011112 000414          BR      8$          :ALL DONE
2421
2422 011114 012704 000022  10$:  MOV     #BDCHK,R4   :CHECKSUM ERROR
2423 011120 004737 012500  CALL    LOG          :
2424 011124 000407          BR      8$          :EXIT
2425
2426 011126 005302          7$:   DEC     R2         :ANY PACKETS LEFT TO CHECK?
2427 011130 001405          BEQ     8$          :NO, ALL DONE
2428 011132 063700 003314  ADD     RBCNT,R0     :YES, POINT TO NEXT PACKET
2429 011136 022121          CMP     (R1)+,(R1)+ :POINT TO NEXT EXPECTED COUNT
2430 011140 022323          CMP     (R3)+,(R3)+ :AND EXPECTED FLAG
2431 011142 000620          BR      1$          :TRY ANOTHER,THEY'RE SMALL
2432 011144 000207          8$:   RETURN          :RETURN

```

```

2435 .SBTTL CHKEND / CHECK SUCCESS AND DETERMINE RETRY STATUS /
2436
2437 :++
2438 : CHKEND - IF RETRYING; DETERMINE IF DATA ERROR OR BAD FLAG BYTE ERROR RETRY.
2439
2440 : IF RETRYING BAD FLAG: RESET RETRY FLAG (SINCE OPERATION IS COMPLETE),
2441 : AND CHECK SUCCESS CODE.
2442 : IF RETRYING DATA ERROR ; CHECK SUCCESS CODE AND IF 0, PRINT RECOVERED, LOG
2443 : SOFT ERROR, END RETRY STATUS. IF NOT 0 AND WAS STILL 'DATA
2444 : CHECK' ERROR - DETERMINE WHETHER TO CONTINUE ANOTHER RETRY OR
2445 : LOG 'UNRECOVERABLE' ERROR.
2446
2447 : IF NOT RETRYING DATA ERROR: CHECK IF 'DATA CHECK' ERROR SUCCESS CODE,
2448 : AND IF SO, START RETRY, ELSE EXIT.
2449 : INPUTS: IMPLIED UNITS DATA BLOCK
2450 : OUTPUTS: RETRY (SYSTAT BIT 1 AND 2), (BIT10 @R5) RESET IF RETRYING.
2451 : - DATA COMARE ERROR (BIT6 @R5) CLEARED.
2452 : - REDUCED/NORMAL GAIN (BIT7 @R5) ADJUSTED
2453 :--
2454
2455 011146 000240 CHKEND:: NOP
2456 011150 010046 PUSH R0 ;R0 --> END PAK
;MOV R0,-(SP)

2457 011152 010446 PUSH R4 ;MOV R4,-(SP)

2458 011154 032737 000006 003304 1$: BIT #BIT1,BIT2,SYSTAT ;RETRYING?
2459 011162 001406 BEQ NOREE ;NO-CHECK NORMALLY
2460 011164 032737 000004 003304 BIT #BIT2,SYSTAT ;IS IT BAD FLAG TYPE?
2461 011172 001454 BEQ CHKERR ;NO(DATA TYPE)
2462 011174 042715 020000 BIC #BIT13,@R5 ;YES, SO IF END PACK THEN RETRY'S COMPLETE
2463 011200 004737 012164 NOREE: CALL CHKSUC ;CHECK SUCCESS CODE
2464 011204 032715 100000 BIT #BIT15,@R5 ;ABORTED?
2465 011210 001402 BEQ 3$ ;NO, CONTINUE
2466 011212 000137 011670 JMP CHKRET ;YES, EXIT
2467 011216 105765 000077 3$: TSTB SUCCS+1(R5) ;NO; HOW'D WE DO?
2468 011222 001013 BNE CHKERR ;NOT SO GOOD.
2469 011224 032715 000100 BIT #BIT6,@R5 ;OK, HOST FIND DATA PAK ERROR?
2470 011230 001002 BNE 2$ ;YES
2471 011232 000137 011670 JMP CHKRET ;NO
2472 011236 012704 000014 2$: MOV #BDCOM,R4 ;YES; JUST BAD DATA-NO DATACHK ERR
2473 011242 004737 012500 CALL LOG ;BAD DATA IN PACKET
2474 011246 000137 011670 JMP CHKRET ;QUIT
2475 011252 032715 001000 CHKERR: BIT #BIT9,@R5 ;BAD SUCCESS; TU DATA CHK ERROR?
2476 011256 001002 BNE 1$ ;YES
2477 011260 000137 011670 JMP CHKRET ;NO. ALL DONE.
2478 011264 052715 002000 1$: BIS #BIT10,@R5 ;YES-START RETRY
2479 011270 012765 000001 000002 MOV #1,RETRY(R5) ;CALL IT 1ST
2480 011276 PRINTX #RTRYN,RETRY(R5) ;** PRINT **

;MOV RETRY(R5),-(SP)
;MOV #RTRYN,-(SP)
;MOV #2,-(SP)
;MOV SP,R0
;TRAP C$PNTX

```

```

2481 011316 062706 000006                                ADD      #6,SP
2482 011322 000562                                ;ALL DONE
2483 011324 004737 012164    CHKREE: CALL  CHKRET      ;CHECK SUCCESS CODE
2484 011330 105765 000077    TSTB   SUCCS+1(R5)    ; SUCCESSFUL YET?
2485 011334 001054    BNE    UNSUC         ;NO, CHECK COUNT
2485 011336                                PRINTX  #RECOV,RETRY(R5)
2485 011336 016546 000002                                MOV      RETRY(R5),-(SP)
2485 011342 012746 011710                                MOV      #RECOV, -(SP)
2485 011346 012746 000002                                MOV      #2, -(SP)
2485 011352 010600                                MOV      SP,R0
2485 011354 104415                                TRAP    C$PNTX
2486 011356 062706 000006                                ADD      #6,SP
2486 011362 105715                                TSTB   (R5)          ;DETERMINE THRESHOLD
2487 011364 100411                                BMI    2$            ;IT'S MODIFIED
2488 011366                                PRINTX  #THRSLO      ;NORMAL
2488 011366 012746 011770                                MOV      #THRSLO, -(SP)
2488 011372 012746 000001                                MOV      #1, -(SP)
2488 011376 010600                                MOV      SP,R0
2488 011400 104415                                TRAP    C$PNTX
2489 011402 062706 000004                                ADD      #4,SP
2489 011406 000410                                BR      3$
2490 011410                                PRINTX  #THRSHI      ;ENHANCED
2490 011410 012746 012016                                MOV      #THRSHI, -(SP)
2490 011414 012746 000001                                MOV      #1, -(SP)
2490 011420 010600                                MOV      SP,R0
2490 011422 104415                                TRAP    C$PNTX
2491 011424 062706 000004                                ADD      #4,SP
2491 011430 032715 000400    3$:  BIT    #BIT8,@R5    ;WRITE OR READ OPERATION?
2492 011434 001003                                BNE    4$            ;WRITE
2493 011436 012704 000002                                MOV      #SFTRD,R4    ;READ
2494 011442 000402                                BR      5$
2495 011444 012704 000004    4$:  MOV      #SFTWR,R4    ;WRITE
2496 011450 004737 012500    5$:  CALL   LOG
2497 011454 005065 000002                                CLR     RETRY(R5)     ;RESTORE TO NORMAL STATE
2498 011460 042715 002200                                BIC    #BIT10!BIT7,@R5 ;NO RETRY, NORM THRESHOLD
2499 011464 000501                                BR      CHKRET        ;QUIT
2500
2501 011466 000240                                UNSUC: NOP
2502 011470 032715 001000                                BIT    #BIT9,@R5     ;RETRYING; SEE IF HARD YET
2503 011474 001015                                BNE    2$            ;TU DATA CHECK ERROR?
2504 011476                                PRINTB  #RETERR      ;YES
2504 011476 012746 012112                                ;NO-'OTHER-ERROR' ERROR
2504 011502 012746 000001                                MOV      #RETERR, -(SP)
2504 011506 010600                                MOV      #1, -(SP)
2504 011510 104414                                MOV      SP,R0
2504 011512 062706 000004                                TRAP    C$PNTB
2505 011516 005065 000002                                ADD      #4,SP
2505 011522 042715 002200                                CLR     RETRY(R5)     ;NO RETRIES
2506 011526 000460                                BIC    #BIT10!BIT7,@R5 ;NO RETRY, NORM THRESHOLD
2507 011530 023765 003326 000002 2$:  BR      CHKRET        ;EXIT
2508 011536 001425                                CMP     MXRTRY,RETRY(R5) ;YES. DID WE GRADUATE TO HARD?
2509 011540 005265 000002                                BEQ    HRD1          ;YES
2510 011544                                INC     RETRY(R5)     ;NO. JUST ANOTHER
2511 011544                                PRINTX  #RTRYN,RETRY(R5) ;PRINT OUT
2511 011544 016546 000002                                MOV      RETRY(R5), -(SP)
2511 011550 012746 012050                                MOV      #RTRYN, -(SP)
2511 011554 012746 000002                                MOV      #2, -(SP)
2511 011560 010600                                MOV      SP,R0

```



```

2550 .SBTTL CHKSUC / INTERPRET SUCCESS CODE /
2551
2552
2553 :++
2554 : CHKSUC - COPY SUCCESS CODE (BYTE) TO SUCCS+1(R5). INTERPRET SUCCESS
2555 : AND IF NOT 0, LOG APPROPRIATE ERROR.
2556 : INPUTS: R0 POINTS TO END PACKET.
2557 : @R5 - UNIT STATUS WORD
2558 : CMDSNT(R5) - COMMAND BYTE
2559
2560 : OUTPUTS: R4 IS ERROR NUMBER IF ERROR.
2561 : SUCCS(R5) UPDATED.
2562 : BIT9 @R5 SET ON DATA CHECK SUCCESS CODE
2563 :--
2564 012164 000240 CHKSUC:: NOP
2565 012166 016065 000002 000076 MOV 2(R0),SUCCS(R5) ;R0-->END PACKET
2566 012174 122760 000000 000003 CMPB #ESOK,3(R0) ;GET SUCCESS BYTE
2567 012202 001535 BEQ 12$ ;COMPLETE SUCCESS-EXIT
2568
2569 012204 122760 000001 000003 CMPB #ESTRY,3(R0) ;OK BUT RETRIES?
2570 012212 001012 BNE 20$ ;NO
2571 012214 126527 000100 000002 CMPB CMDSNT(R5),#RSSRD ;A READ?
2572 012222 001001 BNE 22$ ;NO
2573
2574 012224 000520 BR 10$ ;NO RETRIES IN MAINTENANCE
2575 012226 126527 000100 000003 22$: CMPB CMDSNT(R5),#RSSWR ;A WRITE?
2576 012234 001001 BNE 20$ ;NO
2577 012236 000513 BR 10$ ;LOG IT
2578 012240 122760 177737 000003 20$: CMPB #ESNOMO,3(R0) ;NO MOTOR?
2579 012246 001003 BNE 1$ ;NO
2580 012250 012704 000030 MOV #NOMOT,R4 ;YES-
2581 012254 000506 BR 11$ ;LOG
2582
2583 012256 122760 177757 000003 1$: CMPB #ESCKS,3(R0) ;'DATA CHECK' ERROR?
2584 012264 001003 BNE 2$ ;NO
2585 012266 052715 001000 BIS #BIT9,@R5 ;SET DATA-CHK-ERROR FLAG
2586 012272 000501 BR 12$ ;DONT LOG
2587
2588 012274 126527 000100 000007 2$: CMPB CMDSNT(R5),#RSSSLF ;SELF TEST?
2589 012302 001006 BNE 3$ ;NOPE
2590 012304 105760 000003 TSTB 3(R0) ;YES, NEG. IF ERROR
2591 012310 100072 BPL 12$ ;OK
2592
2593 012312 012704 000044 MOV #SLFER,R4 ;YES-ERROR
2594 012316 000465 BR 11$ ;LOG IT
2595
2596 012320 122760 177740 000003 3$: CMPB #ESSK,3(R0) ;SEEK ERROR?
2597 012326 001005 BNE 4$ ;NO
2598 012330 012704 000024 MOV #SKERR,R4 ;YES-
2599 012334 052705 040000 BIS #BIT14,R5 ;SET 'DOBRK' FLAG
2600 012340 000454 BR 11$ ;LOG
2601
2602 012342 122760 177767 000003 4$: CMPB #ESNCRT,3(R0) ;NO CART?
2603 012350 001003 BNE 5$ ;NO
2604 012352 012704 000054 MOV #NCART,R4 ;YES-
2605 012356 000445 BR 11$ ;LOG
2606
    
```

2607	012360	122760	177720	000003	5\$:	CMPB	#ESCMD,3(R0)	:NO UNDERSTAND HOST?
2608	012366	001003				BNE	6\$:NO
2609	012370	012704	000040			MOV	#CMWDER,R4	:YES-
2610	012374	000436				BR	11\$:LOG
2611								
2612	012376	122760	177770	000003	6\$:	CMPB	#ESNONX,3(R0)	:NON EXISTENT UNIT?
2613	012404	001003				BNE	7\$:NO
2614	012406	012704	000036			MOV	#NOUNIT,R4	:YES-
2615	012412	000427				BR	11\$:LOG
2616								
2617	012414	122760	177765	000003	7\$:	CMPB	#ESWLOC,3(R0)	:WRITE LOCKED?
2618	012422	001003				BNE	8\$:NO
2619	012424	012704	000026			MOV	#WRLOCK,R4	:YES-
2620	012430	000420				BR	11\$:LOG
2621								
2622	012432	122760	177776	000003	8\$:	CMPB	#ESPART,3(R0)	:PARTIAL OP?
2623	012440	001003				BNE	9\$:NO
2624	012442	012704	000034			MOV	#PARTL,R4	:YES-
2625	012446	000411				BR	11\$:LOG
2626								
2627	012450	122760	177711	000003	9\$:	CMPB	#ESREC,3(R0)	:WRONG RECORD?
2628	012456	001003				BNE	10\$:NO
2629	012460	012704	000042			MOV	#RECERR,R4	:YES-
2630	012464	000402				BR	11\$:LOG
2631								
2632	012466	012704	000046		10\$:	MOV	#SUCOTL,R4	:UNDEFINED
2633	012472	004737	012500		11\$:	CALL	LOG	:LOG ERROR
2634	012476	000207			12\$:	RETURN		:RETURN

2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651

.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
:--
    
```

2652 012500 010046

```

LOG::  PUSH  R0
      MOV   R0,-(SP)
    
```

2653 012502 010146

```

      PUSH  R1
      MOV   R1,-(SP)
    
```

2654 012504 010346

```

      PUSH  R3
      MOV   R3,-(SP)
    
```

2655 012506 010446

```

      PUSH  R4
      MOV   R4,-(SP)
    
```

2656
2657 012510 011537 002074
2658 012514 042737 177770 002074
2659 012522 010465 000004
2660 012526 012703 000120
2661 012532 060403
2662 012534 060503
2663 012536 004737 013504
2664 012542 103001
2665 012544 005203
2666 012546 122713 000377
2667 012552 001005
2668 012554

```

      MOV   @R5,L$LUN      ;GET UNIT NUMBER
      BIC   #177770,L$LUN ;MASK IT OFF
      MOV   R4,ABNDX(R5)  ;SAVE INDEX IN CASE OF ABORT MESSAGE
      MOV   #LGOFFST,R3  ;OFFSET TO LOW ORDER BYTE (DRIVE)
      ADD   R4,R3         ;FORM INDEX OF PARAM. TO UPDATE
      ADD   R5,R3         ;FORM ABSOLUTE ADDR. THIS UNIT
      CALL  WHCHDR        ;SEE WHICH DRIVE T'WAS
      BCC   2$           ;WAS DRIVE 0
      INC   R3           ;DRIVE 1: POINT TO UPPER BYTE
      CMPB  #255.,@R3    ;POTENTIAL OVERFLOW POSSIBLE?
      BNE   LOGOK        ;NO
      LOGO: ERRDF 0.,OVRFLO,ERRDES ;YES
    
```

```

TRAP  CSERDF
.WORD 0
.WORD OVRFLO
.WORD ERRDES
    
```

2669 012564 000512
2670 012566 105213
2671 012570 111304
2672 012572 016503 000004
2673 012576 012701 002224
2674 012602 066501 000004
2675 012606 042701 000001
2676 012612 032737 000004 016616
2677 012620 001414

```

LOGOK: BR   ABO          ;ABORT UNIT
      INCB @R3          ;INCREMENT THE ERROR
      MOVB @R3,R4       ;TEMP'LY SAVE IT
      MOV  ABNDX(R5),R3 ;GET INDEX AGAIN
      MOV  #RSNTAB,R1   ;FORM ADRS OF MSG
      ADD  ABNDX(R5),R1 ;LIKE THIS
      BIC  #BIT0,R1     ;INSURE WORD BOUNDARY
      BIT  #EVL,FLGLOC ;EVL SELECTED?
      BEQ  LOGOK2       ;NO-CONT
    
```

2678	012622	123704	002220		CMPB	EVLTHR,R4	:YES,OVER THRESHOLD?		
2679	012626	101011			BHI	LOGOK2	:NO		
2680	012630	010337	012642		MOV	R3,DFTL1+2	:YES,LOAD ERROR #		
2681	012634	011137	012644		MOV	@R1,DFTL1+4	:AND MESSAGE ADDR		
2682	012640			DFTL1:	ERRDF	0,DFTL1,ERRDES	:ERROR		
	012640	104455						TRAP	C\$ERDF
	012642	000000						.WORD	0
	012644	012640						.WORD	DFTL1
	012646	013034						.WORD	ERRDES
2683	012650	000460			BR	ABO	:DROP IT		
2684	012652	120327	000014	LOGOK2:	CMPB	R3,#BDCOM	: 'NEVER FATAL' TYPE?		
2685	012656	103011			BHIS	NTSFT	:NO		
2686	012660	010337	012672		MOV	R3,LOG1+2	:YES, ERROR CODE		
2687	012664	011137	012674		MOV	@R1,LOG1+4	:DESCRIPTION		
2688	012670			LOG1:	ERRSOFT	0.,LOG1,ERRDES			
	012670	104457						TRAP	C\$ERSOFT
	012672	000000						.WORD	0
	012674	012670						.WORD	LOG1
	012676	013034						.WORD	ERRDES
2689	012700	000450			BR	LOGO	:EXIT		
2690									
2691	012702	120327	000026	NTSFT:	CMPB	R3,#WRLOCK	:ONE TRY?		
2692	012706	103411			BLO	MABEE	:NO, MAYBE A MULTIPLE		
2693	012710	010337	012722		MOV	R3,LOG2+2.	:YES		
2694	012714	011137	012724		MOV	@R1,LOG2+4			
2695	012720			LOG2:	ERRHRD	0,LOG2,ERRDES	:PRINT HARD MESSAGE		
	012720	104456						TRAP	C\$ERHRD
	012722	000000						.WORD	0
	012724	012720						.WORD	LOG2
	012726	013034						.WORD	ERRDES
2696	012730	000430			BR	ABO	:DROP UNIT		
2697									
2698	012732	042704	177400	MABEE:	BIC	#177400,R4	:NEGATE SIGN EXTEND		
2699	012736	163704	003316	1\$:	SUB	FTLNM,R4	:SEE IF MULTIPLE OF		
2700	012742	001413			BEQ	HRD	:FTLNM-YES!		
2701	012744	103401			BLO	SFT	:NO		
2702	012746	000773			BR	1\$:NOT THERE YET		
2703									
2704	012750	010337	012762	SFT:	MOV	R3,LOG3+2	:ERROR CODE		
2705	012754	011137	012764		MOV	@R1,LOG3+4	:DESCRIPTION		
2706	012760			LOG3:	ERRSOFT	0,LOG3,ERRDES			
	012760	104457						TRAP	C\$ERSOFT
	012762	000000						.WORD	0
	012764	012760						.WORD	LOG3
	012766	013034						.WORD	ERRDES
2707	012770	000414			BR	LOGO	:EXIT		
2708	012772	010337	013004	HRD:	MOV	R3,LOG3B+2	:HARD ERROR CODE		
2709	012776	011137	013006		MOV	@R1,LOG3B+4	:DESCRIPTION		
2710	013002			LOG3B:	ERRHRD	0,LOG3B,ERRDES			
	013002	104456						TRAP	C\$ERHRD
	013004	000000						.WORD	0
	013006	013002						.WORD	LOG3B
	013010	013034						.WORD	ERRDES
2711									
2712	013012	011500		ABO:	MOV	@R5,R0	:GET UNIT NUMBER		
2713	013014	042700	177770		BIC	#177770,R0	:UN-SIGN EXTEND		
2714	013020				DODU	RC	:USE LOGICAL # TO DROP		

2715	013020	104451						TRAP	CSDODU
	013022		LOGO:	POP	R4		;RESTORE		
	013022	012604					MOV	(SP)+,R4	
2716	013024			POP	R3				
	013024	012603					MOV	(SP)+,R3	
2717	013026			POP	R1				
	013026	012601					MOV	(SP)+,R1	
2718	013030			POP	R0				
	013030	012600					MOV	(SP)+,R0	
2719	013032	000207		RETURN			;RETURN		

```

2722
2723
2724
2725
2726
2727 013034          BGNMSG  ERRDES          ;ERROR DESCRIPTION
      013034
2728 013034          PUSH    R0              ;ERRDES::
      013034 010046          MOV     R0,-(SP)

2729 013036          PUSH    R2              ;
      013036 010246          MOV     R2,-(SP)

2730 013040 005002    CLR     R2              ;PRESET TO DATA TYPE
2731 013042 032715 000020  BIT    #BIT4,@R5      ;WHAT PACK TYPE?
2732 013046 001401    BEQ    2$              ;DATA
2733 013050 005202    INC    R2              ;COMMAND
2734 013052 005046    2$: PRINTB #UNIT,<B,DR(R5)>,R2,<B,SYSTAT+1>
      013052 005046          CLR     -(SP)
      013054 153716 003305  BISB   SYSTAT+1,(SP)
      013060 010246          MOV    R2,-(SP)
      013062 005046          CLR     -(SP)
      013064 156516 000060  BISB   DR(R5),(SP)
      013070 012746 013226  MOV    #UNIT,-(SP)
      013074 012746 000004  MOV    #4,-(SP)
      013100 010600          MOV    SP,R0
      013102 104414          TRAP  C$PNTB
      013104 062706 000012  ADD    #12,SP
2735 013110 016500 000064  MOV    REC(R5),R0      ;RECORD NUMBER
2736 013114 016502 000072  MOV    PATTEN(R5),R2   ;DATA EXPECTED
2737 013120          PRINTB #RECID,R0,<B,CMD$NT(R5)>,<B,R2>,<B,SUCC$+1(R5)>
      013120 005046          CLR     -(SP)
      013122 156516 000077  BISB   SUCC$+1(R5),(SP)
      013126 005046          CLR     -(SP)
      013130 150216          BISB   R2,(SP)
      013132 005046          CLR     -(SP)
      013134 156516 000100  BISB   CMD$NT(R5),(SP)
      013140 010046          MOV    R0,-(SP)
      013142 012746 013306  MOV    #RECID,-(SP)
      013146 012746 000005  MOV    #5,-(SP)
      013152 010600          MOV    SP,R0
      013154 104414          TRAP  C$PNTB
      013156 062706 000014  ADD    #14,SP
2738 013162 005765 000074  TST    DLV(R5)         ;DLV ERROR?
2739 013166 001414    BEQ    3$              ;NO
2740 013170          PRINTB #RECID2,DLV(R5) ;YES-PRINT
      013170 016546 000074  MOV    DLV(R5),-(SP)
      013174 012746 013462  MOV    #RECID2,-(SP)
      013200 012746 000002  MOV    #2,-(SP)
      013204 010600          MOV    SP,R0
      013206 104414          TRAP  C$PNTB
      013210 062706 000006  ADD    #6,SP
2741 013214 005065 000074  3$: CLR    DLV(R5)         ;RESET
2742 013220          POP    R2              ;RESTORE
      013220 012602          MOV    (SP)+,R2
    
```

M RI

```
2743 013222          POP      R0
      013222 012600          MOV      (SP)+,R0
2744 013224          ENDMSG          ;EXIT
      013224
      013224 104423          L10003:
2745 013226          045      101    104  UNIT:: .ASCIZ  /%ADrive# %01%A PAK SENT %01%A FLAG RCVD %03%N/ TRAP  C$MSG
2746          .EVEN
2747 013306          045      101    102  RECID:: .ASCIZ  /%ABLOCK# %04%A COMMAND %02%A EXPCTD %03%A SUCCESS %03%N/
2748          .EVEN
2749 013400          103      101    116  OVRFLO: .ASCIZ  /CAN'T UPDATE ERROR OR STATISTIC:OVERFLOW PENDING/
2750          .EVEN
2751 013462          045      101    040  RECID2: .ASCIZ  /%A RCDB WAS %06%N/
2752          .EVEN
```

```
2755 .SBTTL WHCHDR / SEE WHICH DRIVE IS ACTIVE
2756
2757 :++
2758 : INPUTS: DR(R5)
2759 : OUTPUTS: CARRY=DRIVE (1 OR 0)
2760 :--
2761
2762 WHCHDR:: CLC ;CLEAR CARRY
2763 013504 000241
2764
2765 013506 105765 000060 TSTB DR(R5) ;DR 0?
2766 013512 001401 BEQ 2$ ;YES
2767 013514 000261 SEC ;NO
2768
2769 013516 000207 2$: RETURN ;RETURN
```

```

2772 .SBTTL CHKSUM / FORM THE PACKET CHECKSUM
2773
2774 :++
2775 : THE CHECKSUM IS A 16 BIT CHECKSUM WITH END-AROUND CARRY.
2776 :
2777 : INPUTS: R0 -> (POINTS TO) TOP OF PACKET
2778 :         R1 = # OF BYTES
2779 : OUTPUTS: R0 -> WHERE TO PUT CHECKSUM
2780 :         R1 = CHECKSUM
2781 :--
2782
2783
2784 013520          CHKSUM:: PUSH   R3
      013520 010346          MOV     R3,-(SP)

2785 013522          PUSH   R2
      013522 010246          MOV     R2,-(SP)

2786 013524 042737 000001 003304      BIC   #BIT0,SYSTAT ;'CHECKSUM IS ODD' BIT
2787 013532 032701 000001          BIT   #BIT0,R1      ;AN ODD # OF BYTES?
2788 013536 001403          BEQ   1$          ;NO
2789 013540 052737 000001 003304      BIS   #BIT0,SYSTAT ;YES
2790
2791 013546 006001          1$:   ROR   R1          ;/2 FOR WORDS
2792
2793 013550 005003          2$:   CLR   R3          ;PREP CHECKSUM WORD
2794
2795 013552 062003          3$:   ADD   (R0)+,R3      ;FORM SUM
2796 013554 005503          ADC   R3          ;WITH CARRY
2797 013556 005301          DEC   R1          ;MORE WORDS?
2798 013560 001374          BNE   3$          ;YES
2799
2800 013562 032737 000001 003304      BIT   #BIT0,SYSTAT ;WAS IT ODD
2801 013570 001405          BEQ   4$          ;NO
2802 013572 112002          MOVB  (R0)+,R2      ;YES GET NEXT BYTE
2803 013574 042702 177400          BIC   #177400,R2   ;UN-SIGN EXTEND
2804 013600 060203          ADD   R2,R3        ;ADD IT IN
2805 013602 005503          ADC   R3          ;AND CARRY JUST IN CASE
2806
2807 013604 010301          4$:   MOV   R3,R1      ;RETURN IT IN CORRECT PLACE
2808 013606          POP   R2          ;RESTORE
      013606 012602          MOV   (SP)+,R2
2809 013610          POP   R3
      013610 012603          MOV   (SP)+,R3
2810 013612 000207          RETURN          ;RETURN
  
```

```

2813      .SBTTL  CKCKSM / MODULE TO CHECK THE CHKSUMS
2814
2815      :++
2816      : MAKE SURE THE CHECKSUM RECEIVED = THE CHECKSUM CALCULATED.
2817      : INPUTS:  R4 = THE PACKET BYTE COUNT
2818      :           R0 -> THE PACKET TOP
2819      : OUTPUTS:  CARRY SET IF CHECKSUM CALC'D DOES NOT EQUAL CHECKSUM SENT
2820      :           R0 -> THE PACKET TOP
2821      :--
2822
2823
2824 013614 010146  CKCKSM:: PUSH  R1                MOV    R1,-(SP)

2825 013616 010046          PUSH  R0                ;SAVE
                                MOV    R0,-(SP)

2826 013620 010401          MOV    R4,R1                ;COPY BYTE COUNT TO CORRECT
2827 013622 004737 013520  CALL   CHKSUM              ;REGISTER FOR CHKSUM AND
2828                                     ;FORM CHECKSUM
2829
2830      ;HERE R0 --> XMITTED CHKSUM, R1=CHKSUM CALC'D
2831
2832 013626 122001          CMPB  (R0)+,R1            ;LOWER ORDER CHECK
2833 013630 001005          BNE  2$                ;WRONG
2834
2835 013632 000301          SWAB R1                ;OK-PREP FOR
2836
2837 013634 122001          CMPB  (R0)+,R1            ;HIGH ORDER CHECK
2838 013636 001002          BNE  2$                ;WRONG
2839 013640 000241          CLC                    ;OK-CLEAR SAILING
2840
2841 013642 000401          BR   3$                ;EXIT
2842
2843 013644 000261          2$:  SEC                ;LET ERROR BE KNOWN
2844
2845
2846 013646 012600          3$:  POP  R0                MOV    (SP)+,R0
                                013646 012600          MOV    (SP)+,R1
2847 013650 012601          POP  R1                MOV    (SP)+,R1
2848 013652 000207          RETURN              ;RETURN

```

```

2851 .SBTTL DOBRK / MODULE TO INIT TU58 AND TEST INTERRUPTS
2852
2853
2854 :++
2855 DOBRK - SEND RADIAL SERIAL 'BREAK' TO DEVICE:
2856 - SET 'BREAK' ON INTERFACE.
2857 - SEND 8. NULLS
2858 - CLEAR 'BREAK' ON INTERFACE
2859 - SET VECTORS FOR RCV AND XMIT
2860 - SEND 2 BYTES OF 'INIT'
2861 - RECEIVE 'CONTINUE'
2862 - IF RECEIVE GARBAGE OR TIMEOUT - ERROR
2863 - CLEAR INTERRUPTS AND VECTORS
2864 INPUTS: @R5 BIT14 WAS SET - (SEND BREAK)
2865 OUTPUTS: @R5 BIT14 CLEAR IF SUCCESSFUL INIT.
2866 SYSTAT+1 = RECEIVED BYTE
2867 ERRORS R4 = ERROR CODE:
2868 - SEND NOT READY TIMEOUT (TOSNDB)
2869 - NO RESPONSE
2870 - DLV ERROR
2871 - CAN'T INIT
2872 :--
2873 DOBRK:: CLR      INITWD+1      ;CLEAR BYTE RECEIVE ADDR
2874         CLR      BRKTO       ;CLEAR TIME OUT CONSTANT
2875         BIS      #BIT0,@XMSR(R5) ;SET 'BREAK'
2876         MOV      #RSSNIT,CMSNT(R5) ;SAY WE SENT 'INIT'
2877         BIS      #BIT4,@R5      ;PAK SENT TYPE =COMMAND, SORT OF
2878         MOV      #8.,R4        ;BREAK-IT'S-BACK COUNT=8
2879         BREAK     ;SUPERVISOR TAKE FIVE
2880
2881         ;FOR ^C CHECK, ETC.
2882         TSTB     @XMSR(R5)     ;READY?
2883         BMI      4$           ;YES
2884         DEC      BRKTO       ;NO,TIME OUT?
2885         BNE     1$           ;NO
2886         MOV      #TOSNDB,R4    ;YES, SET ERROR CODE
2887         CALL    LOG          ;LOG IT
2888         BR      3$           ;EXIT
2889         MOV     BRKWD,@XMDB(R5) ;SEND NULL
2890         CLR     BRKTO       ;RESET TIME OUT
2891         DEC     R4          ;MORE NULLS TO SEND?
2892         BNE     1$           ;YES
2893         DEC     @XMSR(R5)     ;NO, CLEAR 'BREAK'
2894         MOV     @RCDB(R5),R0  ;HEAVE 'GARBAGE' 1ST BYTE
2895         SETPRI  #PRI00        ;SET TO INTERRUPT FO SURE
2896
2897         MOV     TRAP     #PRI00,R0
2898         TRAP    C$SPRI
2899
2900         SETVEC  TUVECT(R5),#RCVINT,#PRI07 ;SET VECTO INFO
2901
2902         MOV     #PRI07,-(SP)
2903         MOV     #RCVINT,-(SP)
2904         MOV     TUVECT(R5),-(SP)
2905         MOV     #3,-(SP)
2906         TRAP   C$SVEC
2907         ADD     #10,SP
2908
2909         ADD     #4,TUVECT(R5) ;AND INC TO SND VECTOR
2910         SETVEC TUVECT(R5),#SNDINT,#PRI07;AND SET IT
2911
2912         MOV     #PRI07,-(SP)
    
```

```

014034 012746 014300
014040 016546 000204
014044 012746 000003
014050 104437
014052 062706 000010
2898 01405b 162765 000004 000204 SUB #4,TUVECT(R5) ;RESET VECTOR ADDR.
2899 0140c 005037 014410 CLR BRKTO ;RESET TIME OUT
2900 014070 012704 014406 MOV #INITWD,R4 ;USE ADDR. FOR SNDBYT
2901 014074 010437 014412 MOV R4,BRKPTR ;AND SAVE FOR 'WAIT'
2902 014100 052775 000100 000026 BIS #BIT6,@XMSR(R5) ;ENABLE INTER.
2903 014106 004737 014350 CALL WAIT ;AND ENTER LOOP
2904 014112 005715 TST @R5 ;ABORTED FROM TIME OUT?
2905 014114 1C0446 BMI 3$ ;YES-EXIT
2906
2907 014116 005037 014410 CLR BRKTO ;RESET TIME OUT
2908 014122 012704 014406 MOV #INITWD,R4 ;SEND SECOND INIT
2909 014126 010437 014412 MOV R4,BRKPTR ;SAVE POINTER AGAIN
2910 014132 052775 000100 000026 BIS #BIT6,@XMSR(R5) ;AND THEN ENABLE INT
2911 014140 004737 014350 CALL WAIT ;AND WAIT
2912 014144 005715 TST @R5 ;IF ABORTED
2913 014146 100431 BMI 3$ ;THEN EXIT
2914
2915 014150 012704 014407 MOV #INITWD+1,R4 ;WHERE RESPONSE WILL GO (ADDRESS)
2916 014154 010437 014412 MOV R4,BRKPTR ;AND FOR 'WAIT'
2917 014160 052775 000100 000022 BIS #BIT6,@RCSR(R5) ;ENABLE RECIEVE INT.
2918 014166 004737 014350 CALL WAIT ;GET ANSWER
2919 014172 005715 TST @R5 ;ABORTED?
2920 014174 100416 BMI 3$ ;YES.
2921
2922 014176 123727 014407 000020 CMPB INITWD+1,#RSCONT ;NO, IS IT 'CONTINUE'?
2923 014204 001003 BNE 2$ ;NOPE-ERROR
2924
2925 014206 042715 040000 BIC #BIT14,@R5 ;SUCCESSFUL, CLEAR DOBREAK FLAG
2926 014212 000407 BR 3$ ;EXIT
2927
2928 014214 113737 014407 0033 J5 2$: MOVB INITWD+1,SYSTAT+1 ;SAVE BUM RESPONSE
2929 014222 012704 000032 MOV #CNINIT,R4 ;CAN'T INIT CODE
2930 014226 004737 012500 CALL LOG ;LOG IT
2931 ;SCHEDULER WILL TRY AGAIN IF NOT ABORTED
2932
2933 014232 042775 000100 000026 3$: BIC #BIT6,@XMSR(R5) ;CLEAR INTERRUPTS
2934 014240 042775 000100 000022 BIC #BIT6,@RCSR(R5) ; AND FOR RECIEVE
2935 014246 CLRVEC TUVECT(R5) ;RELEASE RECIEVE VECT.
014246 016500 000204 MOV TRAP TUVECT(R5),R0
014252 104436 TRAP CSCVEC
2936 014254 062765 000004 000204 ADD #4,TUVECT(R5) ;AND GET SEND ADDR.
2937 014262 CLRVEC TUVECT(R5) ;AND RELEASE IT
014262 016500 000204 MOV TRAP TUVECT(R5),R0
014266 104436 TRAP CSCVEC
2938 014270 162765 000004 000204 SUB #4,TUVECT(R5) ;RESTORE POINTER
2939 014276 000207 RETURN ;RETURN

```

```

2942          .SBTTL  INTERRUPT SERVICE ROUTINES AND TIMER
2943
2944 014300    BGNSRV  SNDINT          ;'SEND' INTERRUPT SERVICE:
014300                                           SNDINT::

2945
2946 014300    042775  000100  000026  SNDHND: BIC      #BIT6,@XMSR(R5) ;DISABLE INTERRUPT
2947 014306    112475  000030          MOV      @RCDB(R5),DLV(R5) ;SAVE WORD
2948 014312    ENDSRV          MOV      (R4)+,@XMDB(R5);OUTPUT BYTE
014312                                           L10004:
014312 000002          RTI

2949
2950
2951
2952 014314    BGNSRV  RCVINT          ;'RCV' INTERRUPT SERVICE:
014314                                           RCVINT::

2953
2954 014314    042775  000100  000022  RCVHND: BIC      #BIT6,@RCR(R5) ;DISABLE INTS
2955 014322    017565  000024  000074  MOV      @RCDB(R5),DLV(R5) ;SAVE WORD
2956 014330    116524  000074          MOV      DLV(R5),(R4)+ ;BYTE TO BUFFER
2957 014334    005765  000074          TST      DLV(R5) ;ERROR?
2958 014340    100402          BMI      10$ ;YES
2959 014342    005065  000074          CLR      DLV(R5) ;NO CLEAR ERROR
2960 014346    10$:
2961 014346    ENDSRV          L10005:
014346                                           RTI
014346 000002

2962
2963
2964
2965 014350    000240          WAIT:  NOP          ;WAIT LOOP FOR
2966                                           ;INTERRUPT SERVICING
2967 014352    020437  014412          CMP      R4,BRKPTR ;IF=,THEN NO INTERRUPT
2968 014356    001011          BNE      1$ ;GOT ONE!
2969 014360    104422          BREAK ;SUPERVISOR BREAK
014360                                           TRAP   CSBRK
2970 014362    104422          BREAK ;KILL SOME TIME
014362                                           TRAP   CSBRK
2971 014364    005337  014410          DEC      BRKTO ;TIME OUT?
2972 014370    001367          BNE      WAIT ;NO...CONT.
2973 014372    012704  000050          MOV      #TORCVB,R4 ;YES LOAD ERROR #
2974 014376    004737  012500          CALL    LOG ;LOG IT
2975 014402    000207          1$:  RETURN ;RETURN
2976
2977 014404    000000          BRKWD:  .WORD  0 ;NULL
2978 014406    004          INITWD: .BYTE  RSINIT ;INIT COMMAND
2979 014407    000          .BYTE  0 ;RSCONT IS EXPECTED HERE
2980 014410    000000          BRKTO:  .WORD  0 ;TIME OUT
2981 014412    000000          BRKPTR: .WORD  0 ;POINTER TO INITWD
    
```

```

2984 .SBTTL COMPAR/DATA COMPARISON MODULE
2985
2986
2987 :++
2988 : COMPAR - IF 'COMPARE DATA' SELECTED, COMPARE EACH DATA BYTE OF PACKET
2989 : TO PATTEN(R5). SAVE NUMBER OF BYTES NOT CORRECT. IF NOT
2990 : 0, PRINT SOFT ERROR AND TOTAL # WRONG BYTES. SET 'BAD_DATA_
2991 : IN_PACKET' BIT (BIT6 @R5) FOR HIGHER LEVEL MODULES.
2992 : INPUTS: - (CMPDAT) FLAG TO NOT COMPARE (=1)
2993 : - PKPTR(R5) POINTS TO DATA PACK.
2994 : OUTPUTS: BIT6 @R5 (BAD DATA FLAG) ADJUSTED.
2995 : L$LUN - UNIT NUMBER
2996 : PRNSIZ - SIZE OF PACKET
2997 :--
2998 014414 010046 COMPAR:: PUSH R0 ;COMPARE DATA IS DATA PACKET
      014414 010046 MOV R0,-(SP)
      ;
2999 014416 010446 PUSH R4 ;TO PATTERN WRITTEN
      014416 010446 MOV R4,-(SP)
      ;
3000 014420 010146 PUSH R1 ;USING BYTE COUNT IN PACKET
      014420 010146 MOV R1,-(SP)
      ;
3001 014422 005037 014572 CLR BDBYTS ;CLEAR TOTAL WRONG
3002 014426 016504 000104 MOV PKPTR(R5),R4 ;GET TOP OF PACKET
3003 014432 005737 002214 TST CMPDAT ;COMPARE SELECTED?
3004 014436 001451 BEQ 4$ ;NO-EXIT
3005 014440 005204 INC R4 ;YES, LOCATE COUNT
3006 014442 111401 MOVB @R4,R1 ;GET IT
3007 014444 042701 177400 BIC #177400,R1 ;SIGN-UNEXTEND
3008 ;MUST TEST BYTE-WISE...
3009 014450 005204 INC R4 ;-->FIRST DATA BYTE
3010 014452 126524 000072 1$: CMPB PATTEN(R5),(R4)+ ;DATA-WHAT WAS EXPECTED?
3011 014456 001402 BEQ 2$ ;YES
3012 014460 005237 014572 INC BDBYTS ;NO, INCREMENT TOTAL WRONG
3013 014464 005301 2$: DEC R1 ;MORE LEFT?
3014 014466 001371 BNE 1$ ;YES
3015 014470 005737 014572 TST BDBYTS ;ANY WRONG?
3016 014474 001432 BEQ 4$ ;NO
3017 014476 011537 002074 MOV @R5,L$LUN ;GET UNIT NUMBER
3018 014502 042737 177770 002074 BIC #177770,L$LUN ;MASK IT OFF
3019 014510 ERRSOFT 0.,MSBDA,ERRDES ;YES-PRINT 'BAD DATA IN PACKET' ERROR
      014510 104457 TRAP C$ERSOFT
      014512 000000 .WORD 0
      014514 002336 .WORD MSBDA
      014516 013034 .WORD ERRDES
3020 014520 PRINTB #DESC,BDBYTS
      014520 013746 014572 MOV BDBYTS,-(SP)
      014524 012746 014574 MOV #DESC,-(SP)
      014530 012746 000002 MOV #2,-(SP)
      014534 010600 MOV SP,R0
      014536 104414 TRAP C$PNTB
      014540 062706 000006 ADD #6,SP
3021 014544 052715 000100 BIS #BIT6,@R5 ;LET 'EM KNOW UPSTAIRS-BAD DATA FLAG
  
```

```
3022 014550 012737 000204 003334      MOV    #132.,PRNSIZ  ;SIZE IS ONE DATA PACK
3023 014556 004737 014630              CALL   PRNPAK        ;AND PRINT THE PACKET
3024 014562              4$:      POP    R1           ;RESTORE
      014562 012601              MOV    (SP)+,R1
3025 014564              POP    R4
      014564 012604              MOV    (SP)+,R4
3026 014566              POP    R0
      014566 012600              MOV    (SP)+,R0
3027
3028 014570 000207              RETURN
3029
3030 014572 000000              BDBYTS: .WORD
3031 014574   045   101   124  DESC:  .ASCIZ  /%ATOTAL BAD BYTES= %D3%A.%N/
3032              .EVEN
```

```

3035          .SBTTL PRNPAK/MODULE TO PRINT DATA PACKET
3036
3037          :++
3038          : PRNPAK - IF PRINT DATA PACK_ON_ERROR SELECTED: PRINT EACH BYTE OF PACKET
3039          : TO BY PKPTR(R5).
3040          : INPUTS: PRNSIZ - # OF BYTES IN PACKET.
3041          : OUTPUTS: NONE
3042          :--
3043
3044 014630 000240 PRNPAK:: NOP          ;PRINTS 1 PACKET
3045          ;PKPTR(R5)->TOP OF PACKET
3046          ;PRNSIZ (PASSED)=BYTE COUNT
3047 014632          PUSH R0          MOV R0,-(SP)
      014632 010046
3048 014634          PUSH R4          MOV R4,-(SP)
      014634 010446
3049 014636 105737 002212          TSTB PRBUF          ;PRINT PACKET SELECTED?
3050 014642 001451          BEQ 4$          ;NO
3051 014644 016504 000104          MOV PKPTR(R5),R4      ;YES-GET TOP OF PACK
3052 014650 012737 000020 014774 1$: MOV #16.,LNCNT        ;16 BYTES PER LINE
3053 014656 112437 014776          MOVB (R4)+,PRDAT      ;AVOID SIGN EXTEND
3054 014662          PRINTF #PRFORM,<B,PRDAT> ;PRINT BYTE
      014662 005046          CLR -(SP)
      014664 153716 014776          BISB PRDAT,(SP)
      014670 012746 015000          MOV #PRFORM,-(SP)
      014674 012746 000002          MOV #2,-(SP)
      014700 010600          MOV SP,R0
      014702 104417          TRAP C$PNTF
      014704 062706 000006          ADD #6,SP
3055 014710 005337 003334          DEC PRNSIZ          ;ONE LESS
3056 014714 001414          BEQ 3$          ;NO MORE
3057 014716 005337 014774          DEC LNCNT          ;NEW LINE?
3058 014722 001355          BNE 2$          ;NOT YET
3059 014724          PRINTF #CARLF          ;YES
      014724 012746 015010          MOV #CARLF,-(SP)
      014730 012746 000001          MOV #1,-(SP)
      014734 010600          MOV SP,R0
      014736 104417          TRAP C$PNTF
      014740 062706 000004          ADD #4,SP
3060 014744 000741          BR 1$          ;NEXT LINE
3061 014746          PRINTF #CARLF          ;FINISH UP
      014746 012746 015010          MOV #CARLF,-(SP)
      014752 012746 000001          MOV #1,-(SP)
      014756 010600          MOV SP,R0
      014760 104417          TRAP C$PNTF
      014762 062706 000004          ADD #4,SP
3062 014766          POP R4          MOV (SP)+,R4
      014766 012604
3063 014770          POP R0          MOV (SP)+,R0
      014770 012600
3064 014772 000207          RETURN          ;RETURN

```

3065									
3066	014774	000000				LNCNT:	.WORD		
3067	014776	000000				PRDAT:	.WORD		
3068	015000	045	117	063		PRFORM:	.ASCIZ	/%03%A /	
3069							.EVEN		
3070	015010	045	116	000		CARLF:	.ASCIZ	/%N/	
3071							.EVEN		
3072									
3073	015014								
3074						ENDMOD			

3087
 3088
 3116
 3117
 3118
 3119
 3120
 3121
 3122
 3123

.TITLE MISCELLANEOUS SECTIONS
 .SBTTL REPORT CODING SECTION
 BGNMOD
 :++
 : THE REPORT CODING SECTION CONTAINS THE
 : 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
 :--

3124
 3125
 3126
 3127
 3128
 3129
 3130

015014
 015014
 015014 010046
 015016 010146
 015020 010246
 015022 010346
 015024 010446
 015026 010546

BGNRPT
 PUSH R0
 PUSH R1
 PUSH R2
 PUSH R3
 PUSH R4
 PUSH R5

LSRPT::
 MOV R0,-(SP)
 MOV R1,-(SP)
 MOV R2,-(SP)
 MOV R3,-(SP)
 MOV R4,-(SP)
 MOV R5,-(SP)

3131
 3132
 3133
 3134
 3135
 3136
 3137
 3138

015030
 015030 104422
 015032 012737 003346 015442
 015040
 015040 012746 015444
 015044 012746 000001
 015050 010600
 015052 104416
 015054 062706 000004
 015060
 015060 104422
 015062
 015062 012746 015720
 015066 012746 000001
 015072 010600
 015074 104416
 015076 062706 000004
 015102
 015102 104422
 015104 017705 000332

BREAK
 MOV #BLKTBL,RPTR
 PRINTS #STATHD
 BREAK
 PRINTS #STHD2
 1\$: BREAK
 MOV @RPTR,R5

:GET 1ST DEVICE BLOCK
 :HEADER
 :^C CHECK
 :2ND HEADER
 :^C CHECK
 :GET DEVICE BLOCK

TRAP CSBRK
 MOV #STATHD,-(SP)
 MOV #1,-(SP)
 MOV SP,R0
 TRAP CSPNTS
 ADD #4,SP
 TRAP CSBRK
 MOV #STHD2,-(SP)
 MOV #1,-(SP)
 MOV SP,R0
 TRAP CSPNTS
 ADD #4,SP
 TRAP CSBRK

MI
 TE

MISCELLANEOUS SECTIONS
REPORT CODING SECTION

MACRO M1113 25-SEP-81 10:06 PAGE 89-1

F 7

SEQ 0083

MI
TE

3139	015110	032715	004000	BIT	#BIT11,@R5	:UNIT NOT TESTED?	
3140	015114	001131		BNE	2\$:TRUE, DON'T PRINT STATISTICS	
3141						:OK TO PRINT	
3142	015116	011537	015440	MOV	@R5,RLUN	:SAVE STATUS WORD	
3143	015122	042737	177770	BIC	#177770,RLUN	:MASK UNIT NUM.	
3144	015130	116501	000122	MOVB	SOFTW(R5),R1	:SOFTREAD	
3145	015134	042701	177400	BIC	#177400,R1	:SIGN-UNEXTEND	
3146	015140	116502	000124	MOVB	SOFTW(R5),R2	:SOFT WRITE	
3147	015144	042702	177400	BIC	#177400,R2		
3148	015150	116503	000136	MOVB	HARDR(R5),R3	:HARD READ	
3149	015154	042703	177400	BIC	#177400,R3		
3150	015160	116504	000140	MOVB	HARDW(R5),R4	:HARD WRITE	
3151	015164	042704	177400	BIC	#177400,R4		
3152	015170			PRINTS	#FM0,RLUN	:SUMMARY/UNIT #	
	015170	013746	015440				MOV RLUN,-(SP)
	015174	012746	015556				MOV #FM0,-(SP)
	015200	012746	000002				MOV #2,-(SP)
	015204	010600					MOV SP,R0
	015206	104416					TRAP C\$PNTS
	015210	062706	000006				ADD #6,SP
3153	015214			PRINTS	#FM,#0,WRTNO(R5),RDNO(R5),<B,BDATA(R5)>,R1,R2,R3,R4		
	015214	010446					MOV R4,-(SP)
	015216	010346					MOV R3,-(SP)
	015220	010246					MOV R2,-(SP)
	015222	010146					MOV R1,-(SP)
	015224	005046					CLR -(SP)
	015226	156516	000134				BISB BDATA(R5),(SP)
	015232	016546	000114				MOV RDNO(R5),-(SP)
	015236	016546	000110				MOV WRTNO(R5),-(SP)
	015242	012746	000000				MOV #0,-(SP)
	015246	012746	015574				MOV #FM,-(SP)
	015252	012746	000011				MOV #11,-(SP)
	015256	010600					MOV SP,R0
	015260	104416					TRAP C\$PNTS
	015262	062706	000024				ADD #24,SP
3154	015266	116501	000123	MOVB	SOFTW+1(R5),R1	:SAME	
3155	015272	042701	177400	BIC	#177400,R1	:AS	
3156	015276	116502	000125	MOVB	SOFTW+1(R5),R2	:ABOVE	
3157	015302	042702	177400	BIC	#177400,R2	:THIS	
3158	015306	116503	000137	MOVB	HARDR+1(R5),R3	:TIME	
3159	015312	042703	177400	BIC	#177400,R3	:FOR	
3160	015316	116504	000141	MOVB	HARDW+1(R5),R4	:DRIVE	
3161	015322	042704	177400	BIC	#177400,R4	:ONE	
3162							
3163	015326			PRINTS	#FM,#1,WRTN1(R5),RDN1(R5),<B,BDATA+1(R5)>,R1,R2,R3,R4		
	015326	010446					MOV R4,-(SP)
	015330	010346					MOV R3,-(SP)
	015332	010246					MOV R2,-(SP)
	015334	010146					MOV R1,-(SP)
	015336	005046					CLR -(SP)
	015340	156516	000135				BISB BDATA+1(R5),(SP)
	015344	016546	000116				MOV RDN1(R5),-(SP)
	015350	016546	000112				MOV WRTN1(R5),-(SP)
	015354	012746	000001				MOV #1,-(SP)
	015360	012746	015574				MOV #FM,-(SP)
	015364	012746	000011				MOV #11,-(SP)
	015370	010600					MOV SP,R0

MISCELLANEOUS SECTIONS
REPORT CODING SECTION

MACRO M1113 25-SEP-81 10:06 PAGE 89-2

G 7

SEQ 0084

									TRAP	CSPNTS
									ADD	#24,SP
3164	015372	104416								
	015374	062706	000024							
3165	015400	023727	015442	003364	2\$:	CMP	RPTR,#LSTDEV			
3166	015406	103005				BHIS	3\$			
3167	015410	062737	000002	015442		ADD	#2,RPTR			
3168	015416	000137	015102			JMP	1\$			
3169										
3170	015422				3\$:	POP	R5			
	015422	012605						MOV	(SP)+,R5	
3171	015424					POP	R4			
	015424	012604						MOV	(SP)+,R4	
3172	015426					POP	R3			
	015426	012603						MOV	(SP)+,R3	
3173	015430					POP	R2			
	015430	012602						MOV	(SP)+,R2	
3174	015432					POP	R1			
	015432	012601						MOV	(SP)+,R1	
3175	015434					POP	R0			
	015434	012600						MOV	(SP)+,R0	
3176	015436					ENDRPT				
	015436									
	015436	104425							L10006:	
3177	015440	000000				RLUN:	.WORD		TRAP	CSRPT
3178	015442	000000				RPTR:	.WORD			
3179										
3180	015444	045	116	045		STATHD:	.ASCII /%N% DR BLKS WR BLKS RD BDPAK /			
3181	015512	104	103	110			.ASCIZ @DCHK/RD DCHK/WR DCHK/RD DCHK/WR%N@			
3182							.EVEN			
3183	015556	045	101	125		FM0:	.ASCIZ /%AUNIT %D1%N/			
3184							.EVEN			
3185										
3186	015574	045	101	040		FM:	.ASCII /%A %D1% %D5% %D5% %D3% /			
3187	015650	045	104	063			.ASCIZ /%D3% %D3% %D3% %D3% %N/			
3188							.EVEN			
3189	015720	045	101	040		STHD2:	.ASCII /%A			
3190	015765	122	105	103			.ASCIZ /RECOV RECOV UNRECOV UNRECOV%N/			
3191							.EVEN			
3192	016030					ENDMOD				

MI
TE

```

3195          .SBTTL  INITIALIZE SECTION
3196
3197          :++
3198          : THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3199          : AT THE BEGINNING OF EACH PASS.
3200          :--
3201
3202 016030          BGNINIT
3203 016030          LSINIT::
3204 016030 000240          INIT:  NOP
3205 016032 105037 016612          CLR      STRT          ;FOR STATS CLEAR
3206 016036 005037 003340          CLR      TEST8        ;***** CLR TST 8 FLAG
3207 016042          READEF  #EF.START      ;START COMMAND?
3208 016042 012700 000040          MOV      #EF.START,R0
3209 016046 104447          TRAP     CSREFG
3210 016050          BNCOMPLETE INIT2      ;NO
3211 016050 103002          BCC      INIT2
3212 016052 005237 016612          INC      STRT          ;YES, SET START FLAG
3213 016056 012737 003346 003310  INIT2:  MOV      #BLKTBL,DEVPTR ;SET ALL UNITS ABORTED:
3214 016064 005004          CLR      R4            ;UNIT NUMBER
3215 016066 017705 165216          1$:  MOV      @DEVPTR,R5     ;GET POINTER
3216 016072 010415          MOV      R4,@R5        ;INSERT UNIT #
3217 016074 052715 100000          BIS      #BIT15,@R5   ;SET ABORTED
3218 016100 052715 004000          BIS      #BIT11,@R5   ;SET UNIT NOT TESTED
3219 016104 006304          ASL      R4            ;*2 FOR LOOK-UP
3220 016106 016465 026152 000102  MOV      BUFTBL(R4),RCVBUF(R5) ;SETUP POINTER TO UNIT'S BUFFER
3221 016114 006204          ASR      R4            ;CORRECT BACK TO UNIT #
3222 016116 023727 003310 003364  CMP      DEVPTR,#LSTDEV ;LAST DEVICE DONE?
3223 016124 103005          BHIS     CHECK        ;YES
3224 016126 062737 000002 003310  ADD      #2,DEVPTR     ;NO-GET
3225 016134 005204          INC      R4            ;NEXT DEVICE AND
3226 016136 000753          BR       1$          ;SERVICE
3227 016140 022737 000010 002012  CHECK:  CMP      #8.,LSUNIT   ;MAKE SURE NOT
3228 016146 103005          BHIS     GETHRD      ;TOO MANY UNITS
3229 016150          ERRSF  101.,TOMANY ;TOMANY-REQUEST ^C
3230 016150 104454          TRAP     CSERSF
3231 016152 000145          .WORD  101
3232 016154 016530          .WORD  TOMANY
3233 016156 000000          .WORD  0
3234 016160          DOCLN          ;EXIT
3235 016160 104444          TRAP     CSDCLN
3236 016162 012737 003346 003310  GETHRD: MOV      #BLKTBL,DEVPTR ;INIT TABLE POINTER
3237 016170 005004          CLR      R4            ;CLEAR DEVICE COUNTER
3238 016172 017705 165112          1$:  MOV      @DEVPTR,R5     ;GET STATUS WORD
3239 016176 010437 002074          MOV      R4,LSLUN     ;UNIT NUM. IN CASE ERROR
3240 016202          GPHARD  R4,R2        ;GET HARD INFO
3241 016202 010400          MOV      R4,R0
3242 016204 104442          TRAP     CS$GPHRD
3243 016206 010002          MOV      R0,R2
3244 016210          BNCOMPLETE 3$
3245 016210 103111          BCC     3$
3246 016212 042715 004000          BIC      #BIT11,@R5   ;UNIT IS TESTED!
3247 016216 012203          MOV      (R2)+,R3     ;R3=CSR
3248 016220 012265 000204          MOV      (R2)+,TUVECT(R5) ;GET VECTOR ADDRESS

```

MISCELLANEOUS SECTIONS
INITIALIZE SECTION

MACRO M1113 25-SEP-81 10:06 PAGE 91-1

I 7

SEQ 0086

3239	016224	112265	000061		MOVB	(R2)+,DR+1(R5)	:SAVE UNIT SUMMARY		
3240	016230	005202			INC	R2	:GET TO WORD BOUND		
3241	016232	012237	016614		MOV	(R2)+,PDTFLG	:AND GET PDT FLAG		
3242	016236	052715	040000		BIS	#BIT14,@R5	:SET SEND BREAK FLAG		
3243	016242	032765	000400	000060	BIT	#BIT8,DR(R5)	:DRIVE 0?		
3244	016250	001011			BNE	13\$:YES		
3245	016252	032765	001000	000060	BIT	#BIT9,DR(R5)	:DRIVE 1?		
3246	016260	001005			BNE	13\$:OK		
3247	016262				ERRSF	102.,NODRVS	:NEITHER?!		
	016262	104454						TRAP	C\$ERSF
	016264	000146						.WORD	102
	016266	016560						.WORD	NODRVS
	016270	000000						.WORD	0
3248	016272				DOCLN		:EXIT		
	016272	104444						TRAP	C\$DCLN
3249									
3250	016274	105737	016612		13\$:	TSTB	STRT	:START COMMAND?	
3251	016300	001412				BEQ	14\$:NO, DONT CLEAR	
3252								:YES-CLEAR STATS	
3253	016302	012702	000202		MOV	#BLKEND,R2	:R2-->END OF STATS		
3254	016306	012701	000110		MOV	#WRTNO,R1	:FORM ADDRESS OF START:		
3255	016312	060501			ADD	R5,R1	:R1-->START OF STATS.		
3256	016314	162702	000110		SUB	#WRTNO,R2	:FORM # TO CLEAR		
3257									
3258	016320	105021			2\$:	CLRB	(R1)+	:CLEAR 'EM	
3259	016322	005302				DEC	R2	:MORE?	
3260	016324	001375				BNE	2\$:YES	
3261	016326	042715	100000		14\$:	BIC	#BIT15,@R5	:SET NOT ABORTED	
3262	016332	010365	000022			MOV	R3,RCSR(R5)	:GET DEVICE REGISTERS:	
3263	016336	062703	000002			ADD	#2,R3		
3264	016342	010365	000024			MOV	R3,RCDB(R5)		
3265	016346	062703	000002			ADD	#2,R3		
3266	016352	010365	000026			MOV	R3,XMSR(R5)		
3267	016356	062703	000002			ADD	#2,R3		
3268	016362	105737	016614			TSTB	PDTFLG	:UNIT A PDT?	
3269	016366	001402				BEQ	4\$:NO	
3270	016370	162703	000004			SUB	#4,R3	:YES...RCDB=XMDB	
3271	016374	010365	000030		4\$:	MOV	R3,XMDB(R5)		
3272	016400	005065	000072			CLR	PATTEN(R5)	:ZERO DATA PATTERN	
3273	016404	005065	000002			CLR	RETRY(R5)	:NO RETRIES	
3274	016410	005065	000064			CLR	REC(R5)	:NO RECORD	
3275	016414	005065	000076			CLR	SUCCS(R5)	:NO SUCCESS	
3276	016420	005065	000074			CLR	DLV(R5)	:NO DLV ERROR	
3277	016424	005065	000210			CLR	MRSP(R5)	:***** CLR MRSP INDICATOR	
3278	016430	005037	003336			CLR	ALLGON	:OK TO PRINT STATISTICS	
3279	016434	062737	000002	003310	3\$:	ADD	#2,DEVPTR	:-->NEXT DEVICE	
3280	016442	005204				INC	R4	:INCREMENT UNIT NUMBER	
3281	016444	020437	002012			CMP	R4,L\$UNIT	:MORE UNITS?	
3282	016450	001250				BNE	1\$:YES, GP HARD THE NEXT	
3283									
3284	016452	005037	003304			CLR	SYSTAT	:SYSTEM STATUS WORD	
3285	016456					RFLAGS	FLGLOC	:GET USER FLAGS	
	016456	104421							TRAP
	016460	010037	016616					MOV	C\$RFLA
3286	016464	005037	003330		5\$:	CLR	BLKER	:NO ERROR	
3287	016470	013737	002206	003306	SETLEN:	MOV	LENGTH,TAPLEN	:GET # OF RECORDS	
3288	016476	006237	003306			ASR	TAPLEN	:GET # BLOCKS PER TRACK	

MI
TE

MISCELLANEOUS SECTIONS
INITIALIZE SECTION

MACRO M1113 25-SEP-81 10:06 PAGE 91-2

J 7

SEQ 0087

3289 016502 012737 000200 003332 MOV #200,SECREC ;PRESET SECOND START AT 200
3290 016510 022737 000200 003306 CMP #200,TAPLEN ;# BLKS > 128.?
3291 016516 101003 BHI 3\$;NO-SWITCH TRACKS 2ND PASS
3292 016520 012737 000400 003332 MOV #400,SECREC ;YES-START AT 400

3302
3314
3315 016526 3\$: ENDINIT

L10007: TRAP CSINIT

016526
016526 104411

3316
3317
3318 016530 124 117 117 TOMANY: .ASCIZ /TOO MANY UNITS MAX.=8 /
3319 .EVEN
3320 016560 123 105 114 NODRVS: .ASCIZ /SELECT AT LEAST 1 DRIVE /
3321 .EVEN
3322 016612 000000 STRT:: .WORD
3323 016614 000000 PDFTLG:: .WORD ;TU58 IS IN PDT
3324 016616 000000 FLGLOC:: .WORD ;USER FLAGS

MI
TE

```

3327
3328
3329
3330
3331
3332 016620
016620
3333 016620 000240
3334 016622
016622 012746 000340
016626 012746 016730
016632 012746 000004
016636 012746 000003
016642 104437
016644 062706 000010
3335 016650 012737 003346 016726
3336 016656 017705 000044
3337 016662 032715 104000
3338 016666 100403
3339 016670 005775 000022
3340 016674 000240
3341 016676 023727 016726 003364
3342 016704 103004
3343 016706 062737 000002 016726
3344 016714 000760
3345 016716
016716 012700 000004
016722 104436
3346 016724
016724
016724 104461
016726 000000
3347
3348
3349
3350
3351
3352
3353
3354 016730
016730 012746 016762
016734 012746 000001
016740 010600
016742 104417
016744 062706 000004
3355 016750 011500
3356 016752 042700 177770
3357 016756
016756 104451
3358 016760 000002
3359 016762 045 101 101

:++
: 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
LSAUTO::
NOP ;AUTO DROP ROUTINE
SETVEC #4,#TRPHND,#PRI07 ;GET BUS TRAP VEC.
MOV #PRI07,-(SP)
MOV #TRPHND,-(SP)
MOV #4,-(SP)
MOV #3,-(SP)
TRAP C$SVEC
ADD #10,SP

1$: MOV #BLKTBL,TRPPTR ;GET TOP OF DATA BLOCK TABLE
MOV @TRPPTR,R5 ;GET DATA BLOCK
BIT #BIT15!BIT11,@R5 ;NOT TESTED OR ABORTED?
BMI 2$ ;YES
TST @RCSR(R5) ;NO-VALID ADDRESS?
NOP ;YES...(TRAP IF NOT)
2$: CMP TRPPTR,#LSTDEV ;MORE TO TRY?
BHIS 3$ ;NO
ADD #2,TRPPTR ;ON TO NEXT
BR 1$ ;GET IT
3$: CLRVEC #4 ;RESTORE

MOV #4,R0
TRAP C$CVEC

L10010: TRAP C$AUTO

TRPPTR: .WORD

;ILLEGAL ADDRESS TRAP HANDLER:
TRPHND: PRINTF #MSAUTO ;SAY "AUTO DROPPED"
MOV #MSAUTO,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #4,SP

MOV @R5,R0 ;GET UNIT #
BIC #177770,R0 ;MASK IT OFF
DODU R0 ;DROP HIM

TRAP C$DODU

RTI
MSAUTO: .ASCIZ /%AAUTO DROP: %N/
    
```

3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3382
3394
3395

017002
017002
017002 005737 003336
017006 001004
017010 005737 002210
017014 001401
017016
017016 104424

017020
017020
017020 104412

.SBTTL CLEANUP CODING SECTION

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

BGNCLN

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

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

TRAP C\$DRPT

1\$: ENDCLN

L10011:
TRAP C\$CLEAN

```

3398          .SBTTL  DROP UNIT SECTION
3399
3400          :++
3401          : THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3402          : TO NO LONGER BE TESTED.
3403          :--
3404
3405 017022          BGNDU
3406 017022
3407 017022 010046          ;RO=UNIT NUMBER
3408 017022 010046          ;SAVE IT
3409 017022 010046          MOV      R0,-(SP)
3410 017022 010046          LSDU::
3411 017022 010046
3412 017024 010546          PUSH     R5
3413 017024 010546          ;SAVE PRESENT UNIT POINTER
3414 017024 010546          MOV      R5,-(SP)
3415 017026 004737 017066          CALL     GETR5
3416 017026 004737 017066          ;GET POINTER TO UNIT
3417 017032 052715 100000          BIS     #BIT15,@R5
3418 017032 052715 100000          ;SET ABORTED
3419 017036 012605          POP      R5
3420 017036 012605          ;RESTORE PRESENT UNIT POINTER
3421 017036 012605          MOV      (SP)+,R5
3422 017040 012600          POP      R0
3423 017040 012600          ;RETRIEVE UNIT NUMBER
3424 017040 012600          MOV      (SP)+,R0
3425 017042          PRINTF #ABOMSG,R0
3426 017042 010046
3427 017044 012746 017120          MOV      R0,-(SP)
3428 017050 012746 000002          MOV      #ABOMSG,-(SP)
3429 017054 010600          MOV      #2,-(SP)
3430 017056 104417          MOV      SP,R0
3431 017060 062706 000006          TRAP    C$PNTF
3432          ADD      #6,SP
3433 017064          ENDDU
3434 017064 104453
3435 017066 012737 003346 017116          GETR5: MOV     #BLKTBL, PTR
3436 017074 017705 000016          1$:  MOV     @PTR,R5
3437 017100 005300          DEC     R0
3438 017102 100404          BMI     2$
3439 017104 062737 000002 017116          ADD     #2,PTR
3440 017112 000770          BR     1$
3441 017114 000207          2$:  RETURN
3442 017116 000000          PTR: .WORD
3443 017120 045 101 104 ABOMSG: .ASCIZ  /%ADROPPED UNIT %D1%N/
3444          .EVEN
    
```

L10012: TRAP C\$DU

```
3447 .SBTTL ADD UNIT SECTION
3448
3449 :++
3450 : THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
3451 : TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
3452 : TO THE TEST CYCLE.
3453 :--
3454
3455 017146          BGNAU
      017146          L$AU::
3456
3457 ;THE INIT CODE CONTAINS ALL CODE NECESSARY TO ADD A UNIT.
3463
3475
3476
3477
3478 017146          ENDAU
      017146          L10013: TRAP C$AU
      017146 104452
3479
```

```

3538          .SBTTL TEST 1 / DEVICE SELF-DIAGNOSTIC EXECUTION
3539
3540 017150          .NLIST BGNMOD
3541          ME,BEX
3542
3543 017150          BGNTST
3544 017150          TSTID  #TST1
017150 012737 017214 003324          MOV      #TST1,TSTTOP      ;SAVE ADDR OF TEST
017156 004737 006002          CALL     SETUP          ;INIT UNITS TSTPC
017162 004737 005630          CALL     SETDR          ;GET 1ST DRVS.
017166 004737 006050          CALL     RUN            ;DO TEST
017172 004737 005526          CALL     SWAPDR         ;GET NEXT DRVS.
017176 103004          BCC      64$          ;BR NO 2ND DRVS
017200 004737 006002          CALL     SETUP          ;REINIT UNITS TSTPC
017204 004737 006050          CALL     RUN            ;REPEAT TEST
017210          ;DONE
3545 017210          EXIT TST          64$:
017210 104432          TRAP      C$EXIT
017212 000136          .WORD    L10014-.
3546
3547 017214          TST1:  TUSELF
017214 012700 026174          64$:  MOV      #TRBUF,R0      ;FORM COMMAND PACKET
017220 112710 000002          MOVVB   #RSCMND,@R0     ;COMMAND FLAG
017224 112760 000012 000001          MOVVB   #RSMISZ,1(R0)  ;SIZE OF MESSAGE
017232 112760 000007 000002          MOVVB   #RSSSLF,2(R0) ;SELF TEST OPERATION
017240 105060 000003          CLRB   3(R0)          ;NO MODIFIER.
017244 005060 000004          CLR    4(R0)          ;NO DRIVE OR SWITCHES
017250 005060 000006          CLR    6(R0)          ;NO SEQUENCE NUMBER
017254 005060 000010          CLR    8.(R0)        ;NO BYTES
017260 005060 000012          CLR    10.(R0)       ;NO RECORD #
017264 012701 000012          MOV     #RSMISZ,R1    ;GET SIZE
017270 005721          TST    (R1)+          ;+2 FOR CHECKSUM
017272 012765 000016 000070          MOV     #RSSNSZ,SND CNT(R5) ;SIZE TO SEND
017300 004737 013520          CALL   CHKSUM         ;FORM CHECKSUM
017304 010110          MOV     R1,(R0)       ;INSERT INTO PACKET
017306 012765 000002 000034          MOV     #RSEND,XSFLG(R5) ;EXPECT END.
017314 012765 000016 000036          MOV     #RSNDSZ,XSCNT(R5) ;THIS BIG
017322 012765 000001 000032          MOV     #1,XSPKNM(R5) ;AND 1 PACKET
017330 004737 006556          CALL   RSVP           ;SEND
017334 032715 000010          BIT    #BIT3,@R5     ;RETURN TO SCHEDULER
017340 001325          BNE    64$           ;RETRY?(BAD FLAG)
017340          ;YES
3548 017342 005237 003320          INC    DONE
3549 017346 000207          RETURN
3550
3551
3552 017350          ENDTST
017350
017350 104401          L10014: TRAP      C$TST
  
```

```

3555          .SBTTL TEST 2 / SEEK EOT,BOT
3556
3557 017352          BGNTST
3558 017352          TSTID  #TST2          T2::
017352 012737 017416 003324          MOV  #TST2,TSTTOP      ;SAVE ADDR OF TEST
017360 004737 006002          CALL  SETUP          ;INIT UNITS TSTPC
017364 004737 005630          CALL  SE^DR          ;GET 1ST DRVS.
017370 004737 006050          CALL  RUN            ;DO TEST
017374 004737 005526          CALL  SWAPDR         ;GET NEXT DRVS.
017400 103004          BCC   64$           ;BR NO 2ND DRVS
017402 004737 006002          CALL  SETUP          ;REINIT UNITS TSTPC
017406 004737 006050          CALL  RUN            ;REPEAT TEST
017412          64$:          ;DONE
3559 017412          EXIT TST
017412 104432          TRAP  C$EXIT
017414 000206          .WORD  L10015-.
3560
3561
3562 017416 005004          TST2: CLR  R4          ;R4=INDEX INTO RECORD TABLE
3563 017420 016465 017606 000064 1$:  MOV  RECDAT(R4),REC(R5) ;GET THE RECORD
3564
3565 017426          TUSEEK REC(R5),DR(R5) ;SEEK IT
017426 012700 026174          64$:  MOV  #TRBUF,R0        ;-->(POINT TO) XMIT BUFFER
017432 112710 000002          MOVB #RSCMND,@R0     ;FORM COMMAND MESSAGE PACK
017436 112760 000012 000001          MOVB #RSMISZ,1(R0)   ;THIS BIG
017444 112760 000005 000002          MOVB #RSSSEK,2(R0)  ;OP CODE IS SEEK
017452 016560 000064 000012          MOV  REC(R5),10.(R0) ;TO THIS RECORD
017460 116560 000060 000004          MOVB DR(R5),4.(R0)  ;AND WHICH DRIVE
017466 105060 000003          CLRB 3.(R0)         ;NO MODIFIER
017472 105060 000005          CLRB 5.(R0)         ;NO SWITCHES
017476 005060 000006          CLR  6.(R0)         ;NO SEQUENCE #
017502 005060 000010          CLR  8.(R0)         ;NO BYTE COUNT
017506 012701 000012          MOV  #RSMISZ,R1     ;GET COUNT
017512 005721          TST  (R1)+          ;PLUS FLAG + BCNT
                                ;FOR CHECKSUM CALC
017514 004737 013520          CALL  CHKSUM        ;R0-->TOP R1=# OF BYTES
017520 010110          MOV  R1,(R0)        ;INSERT INTO PACKET
                                ;SET UP EXPECTATIONS:
017522 012765 000016 000070          MOV  #RSSNSZ,SNDcnt(R5) ;HOW MANY TO SEND
017530 112765 000002 000034          MOVB #RSCMND,XSFLG(R5) ;EXPECT END PACK
017536 012765 000016 000036          MOV  #RSNDSZ,XSCNT(R5) ;COUNT WITH THIS
017544 012765 000001 000032          MOV  #1.,XSPKMM(R5)  ;EXPECT ONLY 1 PACKET
017552 004737 006556          CALL  RSVP          ;SEND
017556 032715 000010          BIT  #BIT3,@R5     ;AND RETURN TO SCHEDULER
017562 001321          BNE  64$           ;RETRY (FLAG BYTE ERROR)?
                                ;YES
3566
3567 017564 062704 000002          ADD  #2,R4          ;POINT TO NEXT RECORD
3568 017570 026427 017606 177777          CMP  RECDAT(R4),#-1. ;LAST ONE DONE?
3569 017576 001310          BNE  1$            ;NO-LOOP
3570 017600 005237 003320          INC  DONE          ;YES-SET DONE FLAG
3571 017604 000207          RETURN
3572

```

MISCELLANEOUS SECTIONS
TEST 2 / SEEK EOT,BOT

MACRO M1113 25-SEP-81 10:06 PAGE 104-1

D 8

SEQ 0094

3573 017606 000000
3574 017610 000200
3575 017612 000177
3576 017614 000377
3577 017616 000400
3578 017620 177777
3579 017622
017622
017622 104401

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

L10015: TRAP C\$ETST

MI
TE

```

3582          .SBTTL TEST 3 / HIGH ACTIVITY WRITE/READ
3583
3584          : WRITE THEN READ VARYING DATA FOR ALL PHYSICALLY ADJACENT BLOCKS AROUND
3585          : A RECORD, GO HALF-WAY INTO REMAINING TAPE REPEAT UNTIL EOT.
3586
3587          BGNTST
3588          TSTID #TST3
3589          T3::
3590          017624 012737 017670 003324      MOV #TST3,TSTTOP ;SAVE ADDR OF TEST
3591          017624 017624 017624 017624      CALL SETUP ;INIT UNITS TSTPC
3592          017632 004737 006002 017632      CALL SETDR ;GET 1ST DRVS.
3593          017636 004737 005630 017636      CALL RUN ;DO TEST
3594          017642 004737 006050 017642      CALL SWAPDR ;GET NEXT DRVS.
3595          017646 004737 005526 017646      BCC 64$ ;BR NO 2ND DRVS
3596          017652 103004 017652 017652      CALL SETUP ;REINIT UNITS TSTPC
3597          017654 004737 006002 017654      CALL RUN ;REPEAT TEST
3598          017660 004737 006050 017660      ;DONE
3599          017664 017664 017664 017664      64$:
3600          3589 017664 104432 017664 001340      EXIT TST
3601          TRAP C$EXIT
3602          .WORD L10016-.
3603
3604          3590 017670 012765 000100 000066      TST3: MOV #100,TMP(R5) ;INIT TO HALF OF REMAINING
3605          3591 017676 005004 017676 000000      CLR R4 ;FOR INDEX INTO DATA TABLE
3606          3592 017700 005065 000064 017700      CLR REC(R5) ;START AT RECORD 0
3607          3593 017704 016465 021214 000072      1$: MOV TST3PT(R4),PATTEN(R5) ;GET DATA
3608          3594 017712 012700 026174 017712      TUWRIT PATTEN(R5),REC(R5),#512.,DR(R5),#0
3609          3595 017716 112710 000002 017716      72$: MOV #TRBUF,R0 ;MAKE COMMAND PACKET:
3610          3596 017722 112760 000012 000001      MOVB #RSCMND,@R0 ;COMMAND FLAG
3611          3597 017730 112760 000003 000002      MOVB #RSMISZ,1(R0) ;THIS SIZE
3612          3598 017736 112760 000000 000003      MOVB #RSSWR,2(R0) ;INSERT OP CODE-WRITE
3613          3599 017744 116560 000060 000004      MOVB #0,3.(R0) ;VERIFY (1 OR 0)
3614          3600 017752 112760 000020 000005      MOVB DR(R5),4.(R0) ;DRIVE #
3615          3601 017760 005060 000006 000006      MOVB #020,5.(R0) ;MAINTENANCE MODE SWITCH
3616          3602 017764 012760 001000 000010      CLR 6.(R0) ;NO SEQUENCE #
3617          3603 017772 016560 000064 000012      MOV #512.,8.(R0) ;TOTAL COUNT TO WRITE
3618          3604 020000 012701 000012 020000      MOV REC(R5),10.(R0) ;AT RECORD N
3619          3605 020004 005721 020004 020004      MOV #RSMISZ,R1 ;THE PACKET SIZE PLUS+2
3620          3606 020006 012765 000016 000070      TST (R1)+ ;(FLAG AND COUNT) INTO R1
3621          3607 020014 004737 013520 020014      MOV #RSSNSZ,SND CNT(R5) ;LOAD THE SIZE TO SEND
3622          3608 020020 010110 020020 020020      CALL CHKSUM ;R0 --> R1=COUNT
3623          3609 020022 012765 000020 000034      MOV #RSCONT,XSFLG(R5) ;PUT CHKSUM IN PACKET
3624          3610 020030 012765 000001 000036      MOV #1,XSCNT(R5) ;SET UP EXPECTATIONS:
3625          3611 020036 012765 000001 000032      MOV #1,XSPKNM(R5) ;THE FLAG
3626          3612 020044 012702 001000 020044      MOV #512.,R2 ;THE COUNT
3627          3613 020050 004737 006556 020050      CALL RSVP ;THE # PACKETS EXPECTED
3628          3614 020054 032715 000010 020054      BIT #BIT3,@R5 ;GET # OF DATA BYTES
3629          3615 020060 001314 020060 020060      BNE 72$ ;SEND (AND RETURN TO SCHEDULER)
3630          3616 020062 042715 010000 020062      BIC #BIT12,@R5 ;FLAG BYTE ERROR?
3631          3617 020066 012700 026174 020066      MOV #TRBUF,R0 ;YES
3632          3618 020072 020227 000200 020072      CMP R2,#128. ;FLAG FOR LAST PACKET
3633          3619 020076 101004 020076 020076      BHI 65$ ;POINT TO TOP OF BUFFER AGAIN
3634          3620 020100 010201 020100 020100      MOV R2,R1 ;START DATA PACKET(S)
3635          3621 020102 052715 010000 020102      BIS #BIT12,@R5 ;#512. > 128.!
3636          3622 020106 000402 020106 020106      BR 66$ ;#512. < 128.
3637          ;SO LAST PACKET NOW
3638          ;USE REMAINING COUNT
    
```

020110	012701	000200		65\$:	MOV	#128.,R1	:USE 128. BYTES
020114	110160	000001		66\$:	MOVB	R1,1(R0)	:COPY COUNT TO BUFFER
020120	010103				MOV	R1,R3	:R3=COUNTER TO LOAD BUFFER
020122	112710	000001			MOVB	#RSDATA,@R0	:FLAG FIRST
020126	005720				TST	(R0)+	:SKIP COUNT
020130	116520	000072		67\$:	MOVB	PATTEN(R5),(R0)+	:INSERT DATA
020134	005303				DEC	R3	:MORE?
020136	101374				BHI	67\$:YES
020140	012700	026174			MOV	#TRBUF,R0	:-->TOP AGAIN
020144	116001	000001			MOVB	1(R0),R1	:GET COUNT
020150	042701	177400			BIC	#177400,R1	:ZERO SIGN EXTEND
020154	010165	000070			MOV	R1,SNDCNT(R5)	:HOW MANY TO SEND PLUS
020160	062765	000004	000070		ADD	#4,SNDCNT(R5)	:FLAG,COUNT,CHKSUM
020166	062701	000002			ADD	#2,R1	:COMPENSATE FOR FLAG + COUNT
020172	004737	013520			CALL	CHKSUM	:FOR CHECKSUM CALC.
020176	110120				MOVB	R1,(R0)+	:CHKSUM INTO PACKET
020200	000301				SWAB	R1	:EVEN ON AN ODD
020202	110120				MOVB	R1,(R0)+	:BYTE BOUNDARY
020204	032715	010000			BIT	#BIT12,@R5	:LAST DATA PACKET?
020210	001412				BEQ	68\$:NO
020212	012765	000002	000034		MOV	#RSEND,XSFLG(R5)	:YES-EXPECT 'END'
020220	012765	000016	000036		MOV	#RSNDSZ,XSCNT(R5)	:OF THIS SIZE
020226	012765	000001	000032		MOV	#1,XSPKNT(R5)	:AND 1 PACKET
020234	000411				BR	69\$:SEND
020236	012765	000020	000034	68\$:	MOV	#RSCONT,XSFLG(R5)	:(NOT LAST), EXPECT 'CONTINUE'
020244	012765	000001	000036		MOV	#1,XSCNT(R5)	:AND 1 BYTE
020252	012765	000001	000032		MOV	#1,XSPKNT(R5)	:AND 1 PACKET
020260	004737	006556		69\$:	CALL	RSVP	:SEND PACKET
							:AND RETURN TO SCHEDULER
020264	032715	000010			BIT	#BIT3,@R5	:FLAG BYTE RETRY?
020270	001210				BNE	72\$:YES
020272	032715	002000			BIT	#BIT10,@R5	:RETRY DATA ERROR?
020276	001004				BNE	70\$:YES
020300	162702	000200			SUB	#128.,R2	:NO, MORE DATA TO SEND?
020304	101270				BHI	64\$:YES
020306	000502				BR	71\$:NO
020310				70\$:	TURTRY	REC(R5),#512.,DR(R5)	:RETRY HERE
020310	012700	026174		76\$:	MOV	#TRBUF,R0	:FORM CMD PACK:
020314	112710	000002			MOVB	#RSCMD,@R0	:MESSAGE PACK TYPE
020320	112760	000012	000001		MOVB	#RSMSIZ,1(R0)	:THIS BIG
020326	112760	000002	000002		MOVB	#RSSRD,2(R0)	:OP CODE-READ
020334	016560	000064	000012		MOV	REC(R5),10.(R0)	:THIS RECORD
020342	116560	000060	000004		MOVB	DR(R5),4.(R0)	:THIS DRIVE
020350	105060	000003			CLRB	3(R0)	:PRESET NORM THRESHOLD
020354	105715				TSTB	@R5	:REDUCED?
020356	100002				BPL	77\$:NO
020360	105260	000003			INCB	3(R0)	:YES-CHANGE THRESHOLD
020364	012760	001000	000010	77\$:	MOV	#512.,8.(R0)	:# BYTES DESIRED
020372	112760	000020	000005		MOVB	#020,5.(R0)	:MAINTENANCE MODE
020400	005060	000006			CLR	6.(R0)	:NO SEQUENCE #
020404	012701	000012			MOV	#RSMSIZ,R1	:SIZE OF PACKET
020410	005721				TST	(R1)+	:PLUS FLAG+COUNT INTO R1
020412	012765	000016	000070		MOV	#RSSNSZ,SNDCNT(R5)	:SET UP SIZE TO SEND
020420	004737	013520			CALL	CHKSUM	:FORM CHECKSUM R1=COUNT

020424	010110			MOV	R1,(R0)	:INSERT IN PACKET
020426	012701	001000		MOV	#512.,R1	:SET EXPECTATIONS:
020432	012703	000034		MOV	#XSFLG,R3	:CALC # OF DATA PACKETS TO EXPECT
020436	060503			ADD	R5,R3	:OFFSET OF FLAG
020440	005002			CLR	R2	:ABS. ADDR. OF XSFLG
020442	005202			73\$: INC	R2	:PRESET
020444	012723	000001		MOV	#RSDATA,(R3)+	:# PACKETS EXPECTED
020450	012723	000204		MOV	#132.,(R3)+	:LOAD XSFLG
020454	162701	000200		SUB	#128.,R1	:AND EXPECT COUNT
020460	101401			BLOS	75\$:NEG RESULT LAST TIME
020462	000767			BR	73\$:LAST TIME!
020464	005202			75\$: INC	R2	:MORE TO DO
020466	010265	000032		MOV	R2,XSPKNM(R5)	:ADD ONE FOR END PACK
020472	012723	000002		MOV	#RSEND,(R3)+	:SAVE # PACKETS TO EXPECT
020476	012713	000016		MOV	#RSNDSZ,(R3)	:EXPECT AN END
						:THIS BIG-14. BYTES
020502	004737	006556		CALL	RSVP	:SEND
						:AND RETURN TO SCHEDULER
3597 020516				TUREAD	REC(P5),#512.,DR(R5),#0	
020516	012700	026174		82\$: MOV	#TRBUF,R0	:FORM CMND PACK:
020522	112710	000002		MCVB	#RSCMND,@R0	:MESSAGE PACK TYPE
020526	112760	000012	000001	MOVB	#RSMISZ,1(R0)	:THIS BIG
020534	112760	000002	000002	MOVB	#RSSRD,2(R0)	:OP CODE IS READ
020542	016560	000064	000012	MOV	REC(R5),10.(R0)	:THIS RECORD
020550	116560	000060	000004	MOVB	DR(R5),4.(R0)	:THIS DRIVE
020556	112760	000000	000003	MOVB	#0,3.(R0)	:VERIFY
020564	012760	001000	000010	MOV	#512.,8.(R0)	:TOTAL BYTES TO READ
020572	112760	000020	000005	MOVB	#020,5.(R0)	:MAINTENANCE MODE
020600	005060	000006		CLR	6.(R0)	:NO SEQUENCE #
020604	012701	000012		MOV	#RSMISZ,R1	:GET SIZE OF PACKET
020610	005721			TST	(R1)+	:+2 FOR CHECKSUM
020612	012765	000016	000070	MOV	#RSSNSZ,SNDCNT(R5)	:SIZE TO SEND
020620	004737	013520		CALL	CHKSUM	:FORM CHECKSUM R1=COUNT
020624	010110			MOV	R1,(R0)	:INSERT CHECKSUM
020626	012701	001000		MOV	#512.,R1	:SET EXPECTATIONS:
020632	012703	000034		MOV	#XSFLG,R3	:CALC # OF DATA PACKETS TO EXPECT:
020636	060503			ADD	R5,R3	:GET OFFSET
020640	005002			CLR	R2	:ABS. ADDR. OF XSFLG
020642	005202			78\$: INC	R2	:PRESET AS NONE
020644	012723	000001		MOV	#RSDATA,(R3)+	:# PACKETS EXPECTED
020650	012723	000204		MOV	#132.,(R3)+	:LOAD XSFLG
020654	162701	000200		SUB	#128.,R1	:AND EXPECTED COUNT
020660	101401			BLOS	80\$:NEG RESULT LAST TIME
020662	000767			BR	78\$:LAST TIME
020664	005202			80\$: INC	R2	:MORE TO DO
020666	010265	000032		MOV	R2,XSPKNM(R5)	:ADD ONE FOR END PACK
020672	012723	000002		MOV	#RSEND,(R3)+	:SAVE # PACKETS TO EXPECT
020676	012713	000016		MOV	#RSNDSZ,(R3)	:EXPECT AN END ALSO...
020702	004737	006556		CALL	RSVP	:THIS BIG-14. BYTES
						:SEND
						:AND RETURN TO SCHEDULER

020706	032715	002010		81\$:	BIT	#BIT10!BIT3, @R5	:RETRY?
020712	001500				BEQ	79\$:NO.
020714					TURTRY	REC(R5), #512., DR(R5)	:YES
020714	012700	026174		86\$:	MOV	#TRBUF, R0	:FORM CMD PACK:
020720	112710	000002			MOVB	#RSCMD, @R0	:MESSAGE PACK TYPE
020721	112760	000012	000001		MOVB	#RSM5IZ, 1(R0)	:THIS BIG
020732	112760	000002	000002		MOVB	#RSSRD, 2(R0)	:OP CODE-READ
020740	016560	000064	000012		MOV	REC(R5), 10.(R0)	:THIS RECORD
020746	116560	000060	000004		MOVB	DR(R5), 4.(R0)	:THIS DRIVE
020754	105060	000003			CLRB	3(R0)	:PRESET NORM THRESHOLD
020760	105715				TSTB	@R5	:REDUCED?
020762	100002				BPL	87\$:NO
020764	105260	000003			INCB	3(R0)	:YES-CHANGE THRESHOLD
020770	012760	001000	000010	87\$:	MOV	#512., 8.(R0)	:# BYTES DESIRED
020776	112760	000020	000005		MOVB	#020, 5.(R0)	:MAINTENANCE MODE
021004	005060	000006			CLR	6.(R0)	:NO SEQUENCE #
021011	012701	000012			MOV	#RSM5IZ, R1	:SIZE OF PACKET
021014	005721				TST	(R1)+	:PLUS FLAG+COUNT INTO R1
021016	012765	000016	000070		MOV	#RSSNSZ, SDCNT(R5)	:SET UP SIZE TO SEND
021024	004737	013520			CALL	CHKSUM	:FORM CHECKSUM R1=COUNT
021030	010110				MOV	R1, (R0)	:INSERT IN PACKET
021032	012701	001000			MOV	#512., R1	:SET EXPECTATIONS:
021036	012703	000034			MOV	#XSFLG, R3	:CALC # OF DATA PACKETS TO EXPECT
021042	060503				ADD	R5, R3	:OFFSET OF FLAG
021044	005002				CLR	R2	:ABS. ADDR. OF XSFLG
021046	005202			83\$:	INC	R2	:PRESET
021050	012723	000001			MOV	#RSDATA, (R3)+	:# PACKETS EXPECTED
021054	012723	000204			MOV	#132., (R3)+	:LOAD XSFLG
021060	162701	000200			SUB	#128., R1	:AND EXPECT COUNT
021064	101401				BLOS	85\$:NEG RESULT LAST TIME
021066	000767				BR	83\$:LAST TIME!
021070	005202			85\$:	INC	R2	:MORE TO DO
021072	010265	000032			MOV	R2, XSPKNM(R5)	:ADD ONE FOR END PACK
021076	012723	000002			MOV	#RSEND, (R3)+	:SAVE # PACKETS TO EXPECT
021102	012713	000016			MOV	#RSNDSZ, (R3)	:EXPECT AN END
021106	004737	006556			CALL	RSVP	:THIS BIG-14. BYTES
							:SEND
							:AND RETURN TO SCHEDULER
3598	021116	062704	000002		ADD	#2, R4	:POINT TO NEXT DATA
3599	021122	005764	021214		TST	TST3PT(R4)	:END?
3600	021126	001402			BEQ	2\$:YES
3601	021130	000137	017704		JMP	1\$:NO-WRITE, READ NEW DATA
3602	021134	005004		2\$:	CLR	R4	:POINT TO FIRST DATA
3603	021136	062765	000200	000064	ADD	#200, REC(R5)	:BUT NOW USE ADJACENT RECORD
3604	021144	032765	001000	000064	BIT	#1000, REC(R5)	:ALL ADJACENT RECORDS DONE?
3605	021152	001002			BNE	3\$:YES
3606	021154	000137	017704		JMP	1\$:NO-WRITE, READ AT NEW RECORD
3607	021160	162765	001000	000064	3\$:	SUB	#1000, REC(R5)
3608	021166	066565	000066	000064	ADD	TMP(R5), REC(R5)	:RESTORE TO NEXT RECORD
3609	021174	006265	000066		ASR	TMP(R5)	:HALF INTO REST OF TAPE
3610	021200	103402			BCS	4\$:HALF OF HALF FOR NEXT TIME
							:DONE?

```
3611 021202 000137 017704      JMP      1$      ;NO
3612 021206 005237 003320      4$:      INC     DONE  ;YES-SET FLAG
3613 021212 000207          RETURN
3614 021214 000000      TST3PT: .WORD  000000
3615 021216 125252          .WORD  125252
3616 021220 177777          .WORD  177777
3617 021222 052525          .WORD  052525
3618 021224 000000          .WORD  000000
3619
3620
3621 021226          ENDTST
      021226
      021226 104401
```

L10016: TRAP C\$ETST

```

3624
3625          .SBTTL TEST 4 / WRITE SELECTED NUMBER OF BLOCKS
3626
3627 021230          BGNTST
3628 021230          TSTID  #TST4
                                T4::
021230 012737 021274 003324          MOV  #TST4,TSTTOP      ;SAVE ADDR OF TEST
021236 004737 006002          CALL  SETUP          ;INIT UNITS TSTPC
021242 004737 005630          CALL  SETDR          ;GET 1ST DRVS.
021246 004737 006050          CALL  RUN            ;DO TEST
021252 004737 005526          CALL  SWAPDR         ;GET NEXT DRVS.
021256 103004          BCC   64$          ;BR NO 2ND DRVS
021260 004737 006002          CALL  SETUP          ;REINIT UNITS TSTPC
021264 004737 006050          CALL  RUN            ;REPEAT TEST
021270          64$:          ;DONE
3629 021270          EXIT TST
021270 104432          TRAP  C$EXIT
021272 000724          .WORD  L10017-.

3630
3631
3632 021274 005065 000064          TST4: CLR  REC(R5)      ;START AT REC 0
3633 021300 013765 003306 000066          MOV  TAPLEN,TMP(R5) ;GET THE # OF BLOCKS PER TRACK
3634 021306 005065 000062          CLR  TRK(R5)       ;TRK(R5)=1ST OR 2ND PASS COUNTER
3635 021312 016565 000064 000072          1$:  MOV  REC(R5),PATTEN(R5) ;USE RECORD NO. FOR DATA
3636 021320 005737 002216          TST  DRVCHK        ;ADD DR #?
3637 021324 001403          BEQ  10$          ;NO
3638 021326 066565 000060 000072          ADD  DR(R5),PATTEN(R5) ;YES, ADD DRIVE ID
3639 021334          10$:  TUWRIT PATTEN(R5),REC(R5),#512.,DR(R5),#0
                                72$:  MOV  #TRBUF,R0      ;MAKE COMMAND PACKET:
                                MOV  #RSCMND,@R0      ;COMMAND FLAG
                                MOV  #RSMISZ,1(R0)     ;THIS SIZE
                                MOV  #RSSWR,2(R0)      ;INSERT OP CODE-WRITE
                                MOV  #0,3.(R0)         ;VERIFY (1 OR 0)
                                MOV  DR(R5),4.(R0)     ;DRIVE #
                                MOV  #020,5.(R0)      ;MAINTENANCE MODE SWITCH
                                CLR  6.(R0)           ;NO SEQUENCE #
                                MOV  #512,8.(R0)      ;TOTAL COUNT TO WRITE
                                MOV  REC(R5),10.(R0)  ;AT RECORD N
                                MOV  #RSMISZ,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,XSPKNT(R5)    ;THE # PACKETS EXPECTED
                                MOV  #512.,R2         ;GET # OF DATA BYTES
                                CALL  RSVP            ;SEND (AND RETURN TO SCHEDULER)
                                BIT  #BIT3,@R5        ;FLAG BYTE ERROR?
                                BNE  72$             ;YES
                                BIC  #BIT12,@R5       ;FLAG FOR LAST PACKET
                                MOV  #TRBUF,R0        ;POINT TO TOP OF BUFFER AGAIN
                                CMP  R2,#128.         ;START DATA PACKET(S)
                                BHI  65$             ;#512. > 128.!
                                MOV  R2,R1            ;#512.<128.
                                BIS  #BIT12,@R5       ;SO LAST PACKET NOW
                                64$:
021334 012700 026174
021340 112710 000002
021344 112760 000012 000001
021352 112760 000003 000002
021360 112760 000000 000003
021366 116560 000060 000004
021374 112760 000020 000005
021402 005060 000006
021406 012760 001000 000010
021414 016560 000064 000012
021422 012701 000012
021426 005721
021430 012765 000016 000070
021436 004737 013520
021442 010110

021444 012765 000020 000034
021452 012765 000001 000036
021460 012765 000001 000032
021466 012702 001000
021472 004737 006556
021476 032715 000010
021502 001314
021504 042715 010000
021510 012700 026174
021514 020227 000200
021520 101004
021522 010201
021524 052715 010000
    
```

021530	000402				BR	66\$:USE REMAINING COUNT
021532	012701	000200		65\$:	MOV	#128.,R1		:USE 128. BYTES
021536	110160	000001		66\$:	MOVB	R1,1(R0)		:COPY COUNT TO BUFFER
021542	010103				MOV	R1,R3		:R3=COUNTER TO LOAD BUFFER
021544	112710	000001			MOVB	#RSDATA,@R0		:FLAG FIRST
021550	005720				TST	(R0)+		:SKIP COUNT
021552	116520	000072		67\$:	MOVB	PATTEN(R5),(R0)+		:INSERT DATA
021556	005303				DEC	R3		:MORE?
021560	101374				BHI	67\$:YES
021562	012700	026174			MOV	#TRBUF,R0		:-->TOP AGAIN
021566	116001	000001			MOVB	1(R0),R1		:GET COUNT
021572	042701	177400			BIC	#177400,R1		:ZERO SIGN EXTEND
021576	010165	000070			MOV	R1,SND CNT(R5)		:HOW MANY TO SEND PLUS
021602	062765	000004	000070		ADD	#4,SND CNT(R5)		:FLAG,COUNT,CHKSUM
021610	062701	000002			ADD	#2,R1		:COMPENSATE FOR FLAG + COUNT
021614	004737	013520			CALL	CHKSUM		:FOR CHECKSUM CALC.
021620	110120				MOVB	R1,(R0)+		:CHKSUM INTO PACKET
021622	000301				SWAB	R1		:EVEN ON AN ODD
021624	110120				MOVB	R1,(R0)+		:BYTE BOUNDARY
021626	032715	010000			BIT	#BIT12,@R5		:LAST DATA PACKET?
021632	001412				BEQ	68\$:NO
021634	012765	000002	000034		MOV	#RSEND,XSFLG(R5)		:YES-EXPECT 'END'
021642	012765	000016	000036		MOV	#RSNDSZ,XSCNT(R5)		:OF THIS SIZE
021650	012765	000001	000032		MOV	#1,XSPKNT(R5)		:AND 1 PACKET
021656	000411				BR	69\$:SEND
021660	012765	000020	000034	68\$:	MOV	#RSCONT,XSFLG(R5)		:(NOT LAST), EXPECT 'CONTINUE'
021666	012765	000001	000036		MOV	#1,XSCNT(R5)		:AND 1 BYTE
021674	012765	000001	000032		MOV	#1,XSPKNT(R5)		:AND 1 PACKET
021702	004737	006556		69\$:	CALL	RSVP		:SEND PACKET
								:AND RETURN TO SCHEDULER
021706	032715	000010			BIT	#BIT3,@R5		:FLAG BYTE RETRY?
021712	001210				BNE	72\$:YES
021714	032715	002000			BIT	#BIT10,@R5		:RETRY DATA ERROR?
021720	001004				BNE	70\$:YES
021722	162702	000200			SUB	#128.,R2		:NO, MORE DATA TO SEND?
021726	101270				BHI	64\$:YES
021730	000502				BR	71\$:NO
021732				70\$:	TURTRY	REC(R5),#512.,DR(R5)		:RETRY HERE
021732	012700	026174		76\$:	MOV	#TRBUF,R0		:FORM CMD PACK:
021736	112710	000002			MOVB	#RSCMND,@R0		:MESSAGE PACK TYPE
021742	112760	000012	000001		MOVB	#RSMSIZ,1(R0)		:THIS BIG
021750	112760	000002	000002		MOVB	#RSSRD,2(R0)		:OP CODE-READ
021756	016560	000064	000012		MOV	REC(R5),10.(R0)		:THIS RECORD
021764	116560	000060	000004		MOVB	DR(R5),4.(R0)		:THIS DRIVE
021772	105060	000003			CLRB	3(R0)		:PRESET NORM THRESHOLD
021776	105715				TSTB	@R5		:REDUCED?
022000	100002				BPL	77\$:NO
022002	105260	000003			INCB	3(R0)		:YES-CHANGE THRESHOLD
022006	012760	001000	000010	77\$:	MOV	#512.,8.(R0)		:# BYTES DESIRED
022014	112760	000020	000005		MOVB	#020,5.(R0)		:MAINTENANCE MODE
022022	005060	000006			CLR	6.(R0)		:NO SEQUENCE #
022026	012701	000012			MOV	#RSMSIZ,R1		:SIZE OF PACKET
022032	005721				TST	(R1)+		:PLUS FLAG+COUNT INTO R1
022034	012765	000016	000070		MOV	#RSSNSZ,SND CNT(R5)		:SET UP SIZE TO SEND

022042	004737	013520			CALL	CHKSUM		:FORM CHECKSUM R1=COUNT
022046	010110				MOV	R1,(R0)		:INSERT IN PACKET
022050	012701	001000			MOV	#512.,R1		:SET EXPECTATIONS:
022054	012703	000034			MOV	#XSFLG,R3		:CALC # OF DATA PACKETS TO EXPECT
022060	060503				ADD	R5,R3		:OFFSET OF FLAG
022062	005002				CLR	R2		:ABS. ADDR. OF XSFLG
022064	005202				73\$: INC	R2		:PRESET
022066	012723	000001			MOV	#RSDATA,(R3)+		:# PACKETS EXPECTED
022072	012723	000204			MOV	#132.,(R3)+		:LOAD XSFLG
022076	162701	000200			SUB	#128.,R1		:AND EXPECT COUNT
022102	101401				BLOS	75\$:NEG RESULT LAST TIME
022104	000767				BR	73\$:LAST TIME!
022106	005202				75\$: INC	R2		:MORE TO DO
022110	010265	000032			MOV	R2,XSPKRM(R5)		:ADD ONE FOR END PACK
022114	012723	000002			MOV	#RSEND,(R3)+		:SAVE # PACKETS TO EXPECT
022120	012713	000016			MOV	#RSNDSZ,(R3)		:EXPECT AN END
								:THIS BIG-14. BYTES
022124	004737	006556			CALL	RSVP		:SEND
								:AND RETURN TO SCHEDULER
3640	022140	005365	000066		DEC	TMP(R5)		:DO ALL RECORDS FOR THIS TRACK?
3641	022144	001404			BEQ	2\$:YES-GET OTHER TRACK
3642	022146	005265	000064		INC	REC(R5)		:NO-ONTO NEXT RECORD
3643	022152	000137	021312		JMP	1\$:EXECUTE THE WRITE
3644	022156	005765	000062		2\$: TST	TRK(R5)		:DONE 2 TRACKS?
3645	022162	001012			BNE	TST4EX		:YES-EXIT
3646	022164	005265	000062		INC	TRK(R5)		:NO-SET FLAG FOR NEXT PASS
3647	022170	013765	003332	000064	MOV	SECREC,REC(R5)		:GET NEW STARTING BLOCK #
3648	022176	013765	003306	000066	MOV	TAPLEN,TMP(R5)		:RESET # OF BLOCKS
3649	022204	000137	021312		JMP	1\$:AND EXECUTE
3650	022210	005237	003320		TST4EX: INC	DONE		:DONE
3651	022214	000207			RETURN			:RETURN
3652								
3653	022216				ENDTST			
	022216							
	022216	104401						
							L10017:	TRAP C\$ETST

```

3656          .SBTTL TEST 5 / READ SELECTED NUMBER OF BLOCKS
3657
3658 022220          BGNTST
3659 022220          TSTID  #TST5
022220 012737 022264 003324          MOV  #TST5,TSTTOP      ;SAVE ADDR OF TEST
022226 004737 006002          CALL  SETUP          ;INIT UNITS TSTPC
022232 004737 005630          CALL  SETDR         ;GET 1ST DRVS.
022236 004737 006050          CALL  RUN           ;DO TEST
022242 004737 005526          CALL  SWAPDR        ;GET NEXT DRVS.
022246 103004          BCC   64$           ;BR NO 2ND DRVS
022250 004737 006002          CALL  SETUP          ;REINIT UNITS TSTPC
022254 004737 006050          CALL  RUN           ;REPEAT TEST
022260          64$:          ;DONE
3660 022260          EXIT TST
022260 104432          TRAP  C$EXIT
022262 000520          .WORD  L10020-.

3661
3662
3663 022264 005065 000064          TST5: CLR  REC(R5)      ;START AT REC 0
3664 022270 013765 003306 000066          MOV  TAPLEN,TMP(R5) ;GET THE # OF BLOCKS PER TRACK
3665 022276 005065 000062          CLR  TRK(R5)        ;TRK(R5)=1ST OR 2ND PASS
3666 022302 016565 000064 000072          1$:  MOV  REC(R5),PATTEN(R5) ;USE RECORD NO. AS DATA
3667 022310 005737 002216          TST  DRVCHK         ;ADD DR #?
3668 022314 001403          BEQ  10$           ;NO
3669 022316 066565 000060 000072          ADD  DR(R5),PATTEN(R5) ;ADD IN DRIVE ID
3670 022324          10$:  TUREAD REC(R5),#512.,DR(R5),#0

022324 012700 026174          68$:  MOV  #TRBUF,R0      ;FORM CMND PACK:
022330 112710 000002          MOVB #RSCMND,@R0    ;MESSAGE PACK TYPE
022334 112760 000012 000001          MOVB #RSMISZ,1(R0) ;THIS BIG
022342 112760 000002 000002          MOVB #RSSRD,2(R0)  ;OP CODE IS READ
022350 016560 000064 000012          MOV  REC(R5),10.(R0) ;THIS RECORD
022356 116560 000060 000004          MOVB DR(R5),4.(R0)  ;THIS DRIVE
022364 112760 000000 000003          MOVB #0,3.(R0)     ;VERIFY
022372 012760 001000 000010          MOV  #512.,8.(R0)  ;TOTAL BYTES TO READ
022400 112760 000020 000005          MOVB #020,5.(R0)  ;MAINTENANCE MODE
022406 005060 000006          CLR  6.(R0)        ;NO SEQUENCE #
022412 012701 000012          MOV  #RSMISZ,R1    ;GET SIZE OF PACKET
022416 005721          TST  (R1)+         ;+2 FOR CHECKSUM
022420 012765 000016 000070          MOV  #RSSNSZ,SND CNT(R5) ;SIZE TO SEND
022426 004737 013520          CALL CHKSUM        ;FORM CHECKSUM R1=COUNT
022432 010110          MOV  R1,(R0)       ;INSERT CHECKSUM

022434 012701 001000          MOV  #512.,R1      ;SET EXPECTATIONS:
022440 012703 000034          MOV  #XSFLG,R3     ;CALC # OF DATA PACKETS TO EXPECT:
022444 060503          ADD  R5,R3         ;GET OFFSET
022446 005002          CLR  R2            ;ABS. ADDR. OF XSFLG
022450 005202          64$:  INC  R2           ;PRESET AS NONE
022452 012723 000001          MOV  #RSDATA,(R3)+ ;# PACKETS EXPECTED
022456 012723 000204          MOV  #132.,(R3)+  ;LOAD XSFLG
022462 162701 000200          SUB  #128.,R1     ;AND EXPECTED COUNT
022466 101401          BLOS 66$          ;NEG RESULT LAST TIME
022470 000767          BR   64$          ;LAST TIME
022472 005202          66$:  INC  R2           ;MORE TO DO
          ;ADD ONE FOR END PACK
    
```

022474 010265 000032
 022500 012723 000002
 022504 012713 000016
 022510 004737 006556
 022514 032715 002010
 022520 001500
 022522

MOV R2,XSPKRM(R5) ;SAVE # PACKETS TO EXPECT
 MOV #RSEND,(R3)+ ;EXPECT AN END ALSO...
 MOV #RSNDSZ,(R3) ;THIS BIG-14. BYTES
 CALL RSVP ;SEND
 ;AND RETURN TO SCHEDULER
 67\$: BIT #BIT10!BIT3,@R5 ;RETRY?
 BEQ 65\$;NO.
 TURTRY REC(R5),#512.,DR(R5) ;YES

022522 012700 026174
 022526 112710 000002
 022532 112760 000012 000001
 022540 112760 000002 000002
 022546 016560 000064 000012
 022554 116560 000060 000004
 022562 105060 000003
 022566 105715
 022570 100002
 022572 105260 000003
 022576 012760 001000 000010
 022604 112760 000020 000005
 022612 005060 000006
 022616 012701 000012
 022622 005721
 022624 012765 000016 000070

72\$: MOV #TRBUF,R0 ;FORM CMND PACK:
 MOVB #RSCMND,@R0 ;MESSAGE PACK TYPE
 MOVB #RSMSIZ,1(R0) ;THIS BIG
 MOVB #RSSRD,2(R0) ;OP CODE-READ
 MOV REC(R5),10.(R0) ;THIS RECORD
 MOVB DR(R5),4.(R0) ;THIS DRIVE
 CLRB 3(R0) ;PRESET NORM THRESHOLD
 TSTB @R5 ;REDUCED?
 BPL 73\$;NO
 INCB 3(R0) ;YES-CHANGE THRESHOLD
 73\$: MOV #512.,8.(R0) ;# BYTES DESIRED
 MOVB #020,5.(R0) ;MAINTENANCE MODE
 CLR 6.(R0) ;NO SEQUENCE #
 MOV #RSMSIZ,R1 ;SIZE OF PACKET
 TST (R1)+ ;PLUS FLAG+COUNT INTO R1
 MOV #RSSNSZ,SND CNT(R5) ;SET UP SIZE TO SEND

022632 004737 013520
 022636 010110

CALL CHKSUM ;FORM CHECKSUM R1=COUNT
 MOV R1,(R0) ;INSERT IN PACKET

022640 012701 001000

MOV #512.,R1 ;SET EXPECTATIONS:

022644 012703 000034
 022650 060503
 022652 005002
 022654 005202
 022656 012723 000001
 022662 012723 000204
 022666 162701 000200
 022672 101401
 022674 000767
 022676 005202
 022700 010265 000032
 022704 012723 000002
 022710 012713 000016

MOV #XSFLG,R3 ;CALC # OF DATA PACKETS TO EXPECT
 ADD R5,R3 ;OFFSET OF FLAG
 CLR R2 ;ABS. ADDR. OF XSFLG
 69\$: INC R2 ;PRESET
 ;# PACKETS EXPECTED
 MOV #RSDATA,(R3)+ ;LOAD XSFLG
 MOV #132.,(R3)+ ;AND EXPECT COUNT
 SUB #128.,R1 ;NEG RESULT LAST TIME
 BLOS 71\$;LAST TIME!
 BR 69\$;MORE TO DO
 71\$: INC R2 ;ADD ONE FOR END PACK
 MOV R2,XSPKRM(R5) ;SAVE # PACKETS TO EXPECT
 MOV #RSEND,(R3)+ ;EXPECT AN END
 MOV #RSNDSZ,(R3) ;THIS BIG-14. BYTES

022714 004737 006556

CALL RSVP ;SEND
 ;AND RETURN TO SCHEDULER

3671 022724 005365 000066
 3672 022730 001404
 3673 022732 005265 000064
 3674 022736 000137 022302
 3675 022742 005765 000062
 3676 022746 001012
 3677 022750 005265 000062
 3678 022754 013765 003332 000064

2\$: DEC TMP(R5) ;DO ALL RECORDS THIS TRACK?
 BEQ 2\$;YES-GET OTHER TRACK
 INC REC(R5) ;NO-NEXT RECORD
 JMP 1\$;EXECUTE THE READ
 TST TRK(R5) ;DONE 2 TRACKS?
 BNE TST5EX ;YES-EXIT
 INC TRK(R5) ;NO-SET FLAG FOR NEXT PASS
 MOV SEC REC,REC(R5) ;GET NEW STARTING BLOCK #

3679	022762	013765	003306	000066	MOV	TAPLEN, TMP(R5)	:RESET # OF BLOCKS
3680	022770	000137	022302		JMP	1\$:AND EXECUTE
3681	022774	005237	003320		TST5EX: INC	DONE	:DONE
3682	023000	000207			RETURN		:RETURN
3683							
3684	023002				ENDTST		
	023002						
	023002	104401					

L10020: TRAP CSETST

```

3687          .SBTTL TEST 6 / WRITE-VERIFY SELECTED NUMBER OF BLOCKS
3688
3689 023004          BGNTST
023004
3690 023004          TSTID  #TST6
023004 012737 023050 003324          MOV  #TST6,TSTTOP      ;SAVE ADDR OF TEST
023012 004737 006002          CALL  SETUP          ;INIT UNITS TSTPC
023016 004737 005630          CALL  SETDR         ;GET 1ST DRVS.
023022 004737 006050          CALL  RUN           ;DO TEST
023026 004737 005526          CALL  SWAPDR        ;GET NEXT DRVS.
023032 103004          BCC   64$          ;BR NO 2ND DRVS
023034 004737 006002          CALL  SETUP          ;REINIT UNITS TSTPC
023040 004737 006050          CALL  RUN           ;REPEAT TEST
023044          ;DONE
3691 023044          EXIT TST
023044 104432          TRAP  C$EXIT
023046 000724          .WORD  L10021-.

3692
3693
3694 023050 005065 000064          TST6: CLR  REC(R5)      ;START AT REC 0
3695 023054 013765 003306 000066          MOV  TAPLEN,TMP(R5) ;GET THE # OF BLOCKS PER TRACK
3696 023062 005065 000062          CLR  TRK(R5)       ;TRK(R5)=1ST OR 2ND PASS
3697 023066 016565 000064 000072          1$: MOV  REC(R5),PATTEN(R5) ;USE RECORD NO. FOR DATA
3698 023074 005737 002216          TST  DRVCHK        ;ADD DR #?
3699 023100 001403          BEQ  10$          ;NO
3700 023102 066565 000060 000072          ADD  DR(R5),PATTEN(R5) ;ADD DRIVE ID
3701 023110          10$: TUWRIT PATTEN(R5),REC(R5),#512.,DR(R5),#1
023110          72$: MOV  #TRBUF,R0      ;MAKE COMMAND PACKET:
023114 112710 000002          MOVB #RSCMND,@R0   ;COMMAND FLAG
023120 112760 000012 000001          MOVB #RSMISZ,1(R0) ;THIS SIZE
023126 112760 000003 000002          MOVB #RSSWR,2(R0)  ;INSERT OP CODE-WRITE
023134 112760 000001 000003          MOVB #1,3.(R0)     ;VERIFY (1 OR 0)
023142 116560 000060 000004          MOVB DR(R5),4.(R0) ;DRIVE #
023150 112760 000020 000005          MOVB #020,5.(R0)  ;MAINTENANCE MODE SWITCH
023156 005060 000006          CLR  6.(R0)       ;NO SEQUENCE #
023162 012760 001000 000010          MOV  #512,8.(R0)  ;TOTAL COUNT TO WRITE
023170 016560 000064 000012          MOV  REC(R5),10.(R0) ;AT RECORD N
023176 012701 000012          MOV  #RSMISZ,R1   ;THE PACKET SIZE PLUS+2
023202 005721          TST  (R1)+        ;(FLAG AND COUNT) INTO R1
023204 012765 000016 000070          MOV  #RSSNSZ,SND CNT(R5) ;LOAD THE SIZE TO SEND
023212 004737 013520          CALL CHKSUM       ;R0 --> R1=COUNT
023216 010110          MOV  R1,(R0)      ;PUT CHKSUM IN PACKET
                                ;SET UP EXPECTATIONS:
023220 012765 000020 000034          MOV  #RSCONT,XSFLG(R5) ;THE FLAG
023226 012765 000001 000036          MOV  #1,XSCNT(R5)   ;THE COUNT
023234 012765 000001 000032          MOV  #1,XSPKNM(R5) ;THE # PACKETS EXPECTED
023242 012702 001000          MOV  #512.,R2     ;GET # OF DATA BYTES
023246 004737 006556          CALL RSV           ;SEND (AND RETURN TO SCHEDULER)
023252 032715 000010          BIT  #BIT3,@R5    ;FLAG BYTE ERROR?
023256 001314          BNE  72$          ;YES
023260 042715 010000          BIC  #BIT12,@R5   ;FLAG FOR LAST PACKET
023264 012700 026174          64$: MOV  #TRBUF,R0   ;POINT TO TOP OF BUFFER AGAIN
023270 020227 000200          CMP  R2,#128.    ;START DATA PACKET(S)
023274 101004          BHI  65$          ;#512. > 128.!
023276 010201          MOV  R2,R1       ;#512.<128.
023300 052715 010000          BIS  #BIT12,@R5   ;SO LAST PACKET NOW
023304 000402          BR   66$          ;USE REMAINING COUNT
    
```


023622 010110
 023624 012701 001000
 023630 012703 000034
 023634 060503
 023636 005002
 023640 005202
 023642 012723 000001
 023646 012723 000204
 023652 162701 000200
 023656 101401
 023660 000767
 023662 005202
 023664 010265 000032
 023670 012723 000002
 023674 012713 000016
 023700 004737 006556
 3702 023714 005365 000066
 3703 023720 001404
 3704 023722 005265 000064
 3705 023726 000137 023066
 3706 023732 005765 000062
 3707 023736 001012
 3708 023740 005265 000062
 3709 023744 013765 003332 000064
 3710 023752 013765 003306 000066
 3711 023760 000137 023066
 3712 023764 005237 003320
 3713 023770 000207
 3714
 3715 023772
 023772
 023772 104401

```

MOV R1,(R0) ;INSERT IN PACKET
MOV #512.,R1 ;SET EXPECTATIONS:
;CALC # OF DATA PACKETS TO EXPECT
MOV #XSFLG,R3 ;OFFSET OF FLAG
ADD R5,R3 ;ABS. ADDR. OF XSFLG
CLR R2 ;PRESET
73$: INC R2 ;# PACKETS EXPECTED
MOV #RSDATA,(R3)+ ;LOAD XSFLG
MOV #132.,(R3)+ ;AND EXPECT COUNT
SUB #128.,R1 ;NEG RESULT LAST TIME
BLOS 75$ ;LAST TIME!
BR 73$ ;MORE TO DO
75$: INC R2 ;ADD ONE FOR END PACK
MOV R2,XSPKMM(R5) ;SAVE # PACKETS TO EXPECT
MOV #RSEND,(R3)+ ;EXPECT AN END
MOV #RSNDSZ,(R3) ;THIS BIG-14. BYTES

CALL RSVP ;SEND
;AND RETURN TO SCHEDULER

DEC TMP(R5) ;DO ALL RECORDS FOR THIS TRACK?
BEQ 2$ ;YES-GET OTHER TRACK
INC REC(R5) ;NO-NEXT RECORD
JMP 1$ ;EXECUTE THE WRITE
2$: TST TRK(R5) ;DONE 2 TRACKS?
BNE TST6EX ;YES-EXIT
INC TRK(R5) ;NO-SET FLAG FOR NEXT PASS
MOV SECREC,REC(R5) ;GET NEW STARTING BLOCK #
MOV TAPLEN,TMP(R5) ;RESET # OF BLOCKS
JMP 1$ ;AND EXECUTE
TST6EX: INC DONE ;DONE
RETURN ;RETURN

ENDTST
    
```

L10021: TRAP CSETST

```

3718 .SBTTL TEST 7 / READ-REDUCED THRESHOLD SELECTED NUMBER OF BLOCKS
3719
3720 023774 BGNTST
3721 023774 TSTID #TST7
023774 012737 024040 003324 MOV #TST7,TSTTOP ;SAVE ADDR OF TEST
024002 004737 006002 CALL SETUP ;INIT UNITS TSTPC
024006 004737 005630 CALL SETDR ;GET 1ST DRVS.
024012 004737 006050 CALL RUN ;DO TEST
024016 004737 005526 CALL SWAPDR ;GET NEXT DRVS.
024022 103004 BCC 64$ ;BR NO 2ND DRVS
024024 004737 006002 CALL SETUP ;REINIT UNITS TSTPC
024030 004737 006050 CALL RUN ;REPEAT TEST
024034 ;DONE
3722 024034 EXIT TST
024034 104432 TRAP C$EXIT
024036 000520 .WORD L10022-.

3723
3724
3725 024040 005065 000064 TST7: CLR REC(R5) ;START AT REC 0
3726 024044 013765 003306 000066 MOV TAPLEN,TMP(R5) ;GET THE # OF BLOCKS PER TRACK
3727 024052 005065 000062 CLR TRK(R5) ;TRK(R5)=1ST OR 2ND PASS
3728 024056 016565 000064 000072 1$: MOV REC(R5),PATTEN(R5) ;USE RECORD NO. FOR DATA
3729 024064 005737 002216 TST DRVCHK ;ADD DR #?
3730 024070 001403 BEQ 10$ ;NO
3731 024072 066565 000060 000072 ADD DR(R5),PATTEN(R5) ;ADD DRIVE ID
3732 024100 10$: TUREAD REC(R5),#512.,DR(R5),#1

024100 012700 026174 68$: MOV #TRBUF,R0 ;FORM CMND PACK:
024104 112710 000002 MOVB #RSCMND,R0 ;MESSAGE PACK TYPE
024110 112760 000012 000001 MOVB #RSSMSZ,1(R0) ;THIS BIG
024116 112760 000002 000002 MOVB #RSSRD,2(R0) ;OP CODE IS READ
024124 016560 000064 000012 MOV REC(R5),10.(R0) ;THIS RECORD
024132 116560 000060 000004 MOVB DR(R5),4.(R0) ;THIS DRIVE
024140 112760 000001 000003 MOVB #1,3.(R0) ;VERIFY
024146 012760 001000 000010 MOV #512.,8.(R0) ;TOTAL BYTES TO READ
024154 112760 000020 000005 MOVB #020,5.(R0) ;MAINTENANCE MODE
024162 005060 000006 CLR 6.(R0) ;NO SEQUENCE #
024166 012701 000012 MOV #RSSMSZ,R1 ;GET SIZE OF PACKET
024172 005721 TST (R1)+ ;+2 FOR CHECKSUM
024174 012765 000016 000070 MOV #RSSNSZ,SND CNT(R5) ;SIZE TO SEND
024202 004737 013520 CALL CHKSUM ;FORM CHECKSUM R1=COUNT
024206 010110 MOV R1,(R0) ;INSERT CHECKSUM

024210 012701 001000 MOV #512.,R1 ;SET EXPECTATIONS:
024214 012703 000034 MOV #XSFLG,R3 ;CALC # OF DATA PACKETS TO EXPECT:
024220 060503 ADD R5,R3 ;GET OFFSET
024222 005002 CLR R2 ;ABS. ADDR. OF XSFLG
024224 005202 64$: INC R2 ;PRESET AS NONE
024226 012723 000001 MOV #RSSDATA,(R3)+ ;# PACKETS EXPECTED
024232 012723 000204 MOV #132.,(R3)+ ;LOAD XSFLG
024236 162701 000200 SUB #128.,R1 ;AND EXPECTED COUNT
024242 101401 BLOS 66$ ;NEG RESULT LAST TIME
024244 000767 BR 64$ ;LAST TIME
024246 005202 66$: INC R2 ;MORE TO DO
;ADD ONE FOR END PACK
    
```



```

3803                                     ;SET UP EXPECTATIONS:
3804 025044 012765 000016 000070      MOV  #RSSNSZ,SNDcnt(R5) ;HOW MANY TO SEND
3805 025052 112765 000002 000034      MOVb #RSCMND,XSFLG(R5) ;EXPECT END PACK
3806 025060 012765 000016 000036      MOV  #RSNDSZ,XSCNT(R5) ;COUNT WITH THIS
3807 025066 012765 000001 000032      MOV  #1.,XSPKnm(R5) ;EXPECT ONLY 1 PACKET
3808
3809 025074 004737 006556              CALL  RSVP              ;SEND
3810                                     ;AND RETURN TO SCHEDULER
3811
3812 025100 032715 000010              BIT  #BIT3,@R5         ;RETRY (FLAG BYTE ERROR)?
3813 025104 001321                    BNE  65$               ;YES
3814
3815 025106 012700 026174              66$: MOV  #TRBUF,R0       ;-->(POINT TO) XMIT BUFFER
3816 025112 112710 000002              MOVb #RSCMND,@R0      ;FORM COMMAND MESSAGE PACK
3817 025116 112760 000012 000001      MOVb #RSMSIZ,1(R0)   ;THIS BIG
3818 025124 112760 000000 000002      MOVb #RSSNOP,2(R0)  ;OP CODE IS NO-OPERATION
3819 025132 013760 000064 000012      MOV  REC,10.(R0)    ;TO THIS RECORD
3820 025140 105060 000003              CLRB 3.(R0)         ;NO MODIFIER
3821 025144 105060 000004              CLRB 4.(R0)         ;NO DRIVE
3822 025150 112760 000010 000005      MOVb #BIT03,5.(R0) ;SET MRSP SWITCH
3823 025156 005060 000006              CLR  6.(R0)         ;NO SEQUENCE #
3824 025162 005060 000010              CLR  8.(R0)         ;NO BYTE COUNT
3825 025166 012701 000012              MOV  #RSMSIZ,R1     ;GET COUNT
3826 025172 005721                    TST  (R1)+          ;PLUS FLAG + BCNT
3827                                     ;FOR CHECKSUM CALC
3828 025174 004737 013520              CALL  CHKSUM        ;R0-->TOP R1=# OF BYTES
3829 025200 010110                    MOV  R1,(R0)        ;INSERT INTO PACKET
3830                                     ;SET UP EXPECTATIONS:
3831 025202 012765 000016 000070      MOV  #RSSNSZ,SNDcnt(R5) ;HOW MANY TO SEND
3832 025210 112765 000002 000034      MOVb #RSCMND,XSFLG(R5) ;EXPECT END PACK
3833 025216 012765 000016 000036      MOV  #RSNDSZ,XSCNT(R5) ;COUNT WITH THIS
3834 025224 012765 000001 000032      MOV  #1.,XSPKnm(R5) ;EXPECT ONLY 1 PACKET
3835
3836 025232 004737 006556              CALL  RSVP              ;SEND
3837                                     ;AND RETURN TO SCHEDULER
3838
3839 025236 032715 000010              BIT  #BIT3,@R5         ;RETRY (FLAG BYTE ERRGR)?
3840 025242 001321                    BNE  66$               ;YES
3841
3842 025244 005237 003320              INC  DONE
3843 025250 005037 003340              CLR  TEST8
3844
3845 025254 005737 002222              TST  PPSOT8         ;PROTOCOL SUMMARY @ END OF PASS
3846 025260 001477                    BEQ  ENDT8          ;NO
3847 025262 005037 025640              CLR  UNITNO        ;SET UNIT # TO ZERO
3848 025266                    PRINTF #MSAGE1     ;PRINT HEADER
3849 025266 012746 025464              MOV  #MSAGE1,-(SP)
3850 025272 012746 000001              MOV  #1,-(SP)
3851 025276 010600                    MOV  SP,R0
3852 025300 104417                    TRAP C$PNTF
3853 025302 062706 000004              ADD  #4,SP
3849 025306 012737 003346 003310      1$: MOV  #BLKTBL,DEVPtr ;SET ALL UNITS
3850 025314 017705 155770              MOV  @DEVPtr,R5     ;GET POINTER
3851 025320 005765 000000              TST  STATUS(R5)    ;IS UNIT ABORTED
3852 025324 100431                    BMI  3$             ;YES
3853 025326 005765 000210              TST  MRSP(R5)      ;IS UNIT MODIFIED
3854 025332 001413                    BEQ  2$             ;NO
    
```

CSSDDDD

3884
3885
3886
3887
3888 026152 027232
3889 026154 030270
3890 026156 031326
3891 026160 032364
3892 026162 033422
3893 026164 034460
3894 026166 035516
3895 026170 036554
3896
3897
3898
3899
3900
3901 026172 023
3902 026173 023
3903
3904 026174
3905
3906
3907
3908 027232
3909 030270
3910 031326
3911 032364
3912 033422
3913 034460
3914 035516
3915 036554
3916
3917
3918
3919 037612

.SBTTL I/O BUFFER AREAS:

;WHO-GETS-WHAT-SPACE TABLE

BUFTBL: .WORD BUF0
.WORD BUF1
.WORD BUF2
.WORD BUF3
.WORD BUF4
.WORD BUF5
.WORD BUF6
.WORD BUF7

;ONLY 1 TRANSMIT BUFFER NECESSARY:

.BYTE RSXOFF
.BYTE RSXOFF ;SEND XOFF BEFORE EVERY PACKET

TRBUF: .BLKB RCBFSZ

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

ENDMOD

3943
3954
3955
3983
3984 037612
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995 037612

.TITLE PARAMETER CODING
.SBTTL HARDWARE PARAMETER CODING SECTION

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.
:--

BGNHRD

000021

.WORD L10024-L\$HARD/2
L\$HARD::

3996
3997
3998 037614

GPRMA MSG1,0,0,160000,177777,YES

.WORD T\$CODE
.WORD MSG1
.WORD T\$LOLIM
.WORD T\$HILIM

037614 000031
037616 037656
037620 160000
037622 177777

3999 037624
037624 001031
037626 037667
037630 000000
037632 000776

GPRMA MSG1B,2,0,0,776,YES

.WORD T\$CODE
.WORD MSG1B
.WORD T\$LOLIM
.WORD T\$HILIM

4000 037634
037634 003130
037636 037704
037640 000001

GPRML MSG1C,6,1,YES

.WORD T\$CODE
.WORD MSG1C
.WORD 1

4001 037642
037642 002130
037644 037722
037646 000001

GPRML MSG2,4,1,YES

.WORD T\$CODE
.WORD MSG2
.WORD 1

4002 037650
037650 002130
037652 037737
037654 000002

GPRML MSG3,4,2,YES

.WORD T\$CODE
.WORD MSG3
.WORD 2

4003
4009
4010 037656

ENDHRD

.EVEN
L10024:

4011
4012 037656
4013 037667
4014 037704
4015 037722
4016 037737
4017
4018
4019

124 125 065 MSG1: .ASCIZ /TU58 CSR/
126 105 103 MSG1B: .ASCIZ /VECTOR ADDR./
120 104 124 MSG1C: .ASCIZ /PDT INTERFACE/
124 105 123 MSG2: .ASCIZ /TEST DRIVE 0/
124 105 123 MSG3: .ASCIZ /TEST DRIVE 1/
.EVEN

PARAMETER CODING
SYMBOL TABLE

ABNDX = 000004 G
ABO 013012
ABOMSG 017120
ABONM 006530
ADR = 000020 G
ALLGON 003336 G
ASSEMB= 000010
BDATA = 000134 G
BDBYTS 014572
BDCHK = 000022 G
BDCOM = 000014 G
BIT0 = 000001 G
BIT00 = 000001 G
BIT01 = 000002 G
BIT02 = 000004 G
BIT03 = 000010 G
BIT04 = 000020 G
BIT05 = 000040 G
BIT06 = 000100 G
BIT07 = 000200 G
BIT08 = 000400 G
BIT09 = 001000 G
BIT1 = 000002 G
BIT10 = 002000 G
BIT11 = 004000 G
BIT12 = 010000 G
BIT13 = 020000 G
BIT14 = 040000 G
BIT15 = 100000 G
BIT2 = 000004 G
BIT3 = 000010 G
BIT4 = 000020 G
BIT5 = 000040 G
BIT6 = 000100 G
BIT7 = 000200 G
BIT8 = 000400 G
BIT9 = 001000 G
BLKEND= 000202 G
BLKER 003330 G
BLKSIZ= 000212 G
BLKTBL 003346 G
BOE = 000400 G
BRKPTR 014412
BRKTO 014410
BRKWD 014404
BUFTBL 026152
BUF0 027232
BUF1 030270
BUF2 031326
BUF3 032364
BUF4 033422
BUF5 034460
BUF6 035516
BUF7 036554
CARLF 015010
CHECK 016140
CHKANR 010546

CHKANS 010442 G
CHKEND 011146 G
CHKERR 011252
CHKPKS 010552 G
CHKPTR 010550
CHKREE 011324
CHKRET 011670
CHKSUC 012164 G
CHKSUM 013520 G
CHK8 010462
CKCKSM 013614 G
CLRALL 005706 G
CLRBUF 005746 G
CLRPT 006000
CMDSNT= 000100 G
CMNDR= 000040 G
CMPDAT 002214
CNINIT= 000032 G
COMPAR 014414 G
CSNRDY 003342 G
CSRCVB 003344 G
CSAU = 000052
CSAUTO= 000061
CSBRK = 000022
CSBSEG= 000004
CSBSUB= 000002
CSCEFG= 000045
CSCLCK= 000062
CSCLEA= 000012
CSCLOS= 000035
CSCLP1= 000006
CSCVEC= 000036
CSDCLN= 000044
CSDODU= 000051
CSDRPT= 000024
CSDU = 000053
CSEDIT= 000003
CSERDF= 000055
CSERHR= 000056
CSERRO= 000060
CSERSF= 000054
CSERSO= 000057
CSESCA= 000010
CSESEG= 000005
CSESUB= 000003
CSETST= 000001
CSEXIT= 000032
CSGETB= 000026
CSGETW= 000027
CSGMAN= 000043
CSGPHR= 000042
CSGPLO= 000030
CSGPRI= 000040
CSINIT= 000011
CSINLP= 000020
CSMANI= 000050
CSMEM = 000031

C\$MSG = 000023
C\$OPEN= 000034
C\$PNTB= 000014
C\$PNTF= 000017
C\$PNTS= 000016
C\$PNTX= 000015
C\$QIO = 000377
C\$RDBU= 000007
C\$REFG= 000047
C\$RESE= 000033
C\$REVI= 000003
C\$RFLA= 000021
C\$RPT = 000025
C\$SEFG= 000046
C\$SPRI= 000041
C\$SVEC= 000037
C\$TPRI= 000013
DESC 014574
DEVPTR 003310 G
DEVO 003366
DEV1 003600
DEV2 004012
DEV3 004224
DEV4 004436
DEV5 004650
DEV6 005062
DEV7 005274
DFPTBL 002174 G
DFTL1 012640
DIAGMC= 000000
DLV = 000074 G
DOBRK 013654 G
DONE 003320 G
DR = 000060 G
DRVCHK 002216
EF.CON= 000036 G
EF.NEW= 000035 G
EF.PWR= 000034 G
EF.RES= 000037 G
EF.STA= 000040 G
ENDGP8 010114
ENDRSP 007076
ENDTB 025460
ERRDES 013034 G
ESABO = 177720 G
ESCKS = 177757 G
ESCKSM= 177757
ESCMD = 177720 G
ESNCRT= 177767 G
ESNOMO= 177737 G
ESNONX= 177770 G
ESOK = 000000 G
ESPART= 177776 G
ESRD = 177757
ESREC = 177711 G
ESSK = 177740 G
ESSLF = 177777 G

E\$TRY = 000001 G
E\$WLOC= 177765 G
ESWR = 177757
EVL = 000004 G
EVLTHR 002220
EXOFF 007425
EXON 007424
E\$END = 002100
E\$LOAD= 000035
FLGLOC 016616 G
FM 015574
FMO 015556
FTLNM 003316
FSAU = 000015
FSAUTO= 000020
FSBGN = 000040
F\$CLEA= 000007
F\$DU = 000016
F\$END = 000041
F\$HARD= 000004
F\$HW = 000013
F\$INIT= 000006
F\$JMP = 000050
F\$MOD = 000000
F\$MSG = 000011
F\$PROT= 000021
F\$PWR = 000017
F\$RPT = 000012
F\$SEG = 000003
F\$SOFT= 000005
F\$SRV = 000010
F\$SUB = 000002
F\$SW = 000014
F\$TEST= 000001
GBTMP 010436
GBTMP2 010440
GETANS 007150 G
GETHRD 016162
GETPTR 007214
GETR5 017066
GTAGIN 007462
GTBYTE 010212 G
GTDOWN 010052
GTOK 007720
GTPKS1 007216 G
GTPKS8 007426 G
GTPTR 010210
GTUM 007662
G\$CNTO= 000200
G\$DELM= 000372
G\$DISP= 000003
G\$EXCP= 000400
G\$HILI= 000002
G\$LOLI= 000001
G\$NO = 000000
G\$OFFS= 000400
G\$OFFSI= 000376

G\$PRMA= 000001
G\$PRMD= 000002
G\$PRML= 000000
G\$RADA= 000140
G\$RADB= 000000
G\$RADL= 000040
G\$RADO= 000020
G\$XFER= 000004
G\$YES = 000010
HARDR = 000136 G
HARDW = 000140 G
HELP = 000000
HOE = 100000 G
HRD 012772
HRDRD = 000016 G
HRDWR = 000020 G
HRD1 011612
IBE = 010000 G
IDPTR 003322 G
IDU = 000040 G
IER = 020000 G
INIT 016030
INITWD 014406
INIT2 016056
ISR = 000100 G
IXE = 004000 G
ISAU = 000041
ISAUTO= 000041
ISCLN = 000041
ISDU = 000041
ISHRD = 000041
ISINIT= 000041
ISMOD = 000041
ISMSG = 000041
ISPROT= 000040
ISPTAB= 000041
ISPWR = 000041
ISRPT = 000041
ISSEG = 000041
ISSETU= 000041
ISSFT = 000041
ISSRV = 000041
ISSUB = 000041
ISTST = 000041
JSJMP = 000167
LENGTH 002206
LGOFFST= 000120 G
LNCNT 014774
LOE = 040000 G
LOG 012500 G
LOGO 013022
LOGOK 012566
LOGOK2 012652
LOGO 012554
LOG1 012670
LOG2 012720

PARAMETER CODING
SYMBOL TABLE

MACRO M1113 25-SEP-81 10:06 PAGE 126-3

SEQ 0122

TSARGC= 000002
TSCODE= 006130
TSERRN= 000146
TSEXCP= 000000
TSFLAG= 000040
TSFREE= 040436
TSGMAN= 000000
TSHILI= 000376
TSLAST= 000001
TSLOLI= 000001
TSLSYM= 010000
TSLTNO= 000010
TSNEST= 177777
TSNSO = 000000
TSNS1 = 000005
TSPCNT= 000000

TSPTAB= 010027
TSPTHV= 000001
TSPTNU= 000001
TSSAVL= 177777
TSSEGL= 177777
TSSIZE= 000006
TSSUBN= 000000
TSTAGL= 177777
TSTAGN= 010C31
TSTEMP= 000000
TSTEST= 000010
TSTSTM= 177777
TSTSTS= 000001
TSSAU = 010013
TSSAUT= 010010
TSSCLE= 010011

TSSDAT= 010030
TSSDU = 010012
TSSHAR= 010024
TSSHW = 010001
TSSINI= 010007
TSSMSG= 010003
TSSPC = 000001
TSSPRO= 010000
TSSPTA= 010027
TSSRPT= 010006
TSSSOF= 010025
TSSSRV= 010005
TSSSW = 010002
TSTES= 010023
T1 = 017150 G
T1TRY = 000146 G

T2 = 017352 G
T3 = 017624 G
T4 = 021230 G
T4TRY = 000132 G
T5 = 022220 G
T6 = 023004 G
T7 = 023774 G
T8 = 024560 G
UAM = 000200 G
UNIT = 013226 G
UNITNO 025640
UNREC 012070
UNSUC 011466
UNXPCT 007624
WAIT 014350

WHCHDR 013504 G
WRLOCK= 000026 G
WRTNO = 000110 G
WRTN1 = 000112 G
XFNSND 006636
XMDB = 000030 G
XMSR = 000026 G
XSCNT = 000036 G
XSFLG = 000034 G
XSPKMM= 000032 G
XSPTR = 000106 G
XSALWA= 000000
XSFALS= 000040
XSOFFS= 000400
XSTRUE= 000020

. ABS. 040436 000
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 33777 WORDS (132 PAGES)
DYNAMIC MEMORY: 21558 WORDS (82 PAGES)
ELAPSED TIME: 00:06:01
CZTUUC.BIN/EN:AMA:ABS,CZTUUC/CR/SP=LB1:[1,1]SVC/MLB,SY:[203,375]CZTUUC.MAC

CSS

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
ABNDX	=	000004 G	#16-888 *72-2659 72-2672 72-2674
ABO	=	013012	72-2669 72-2683 72-2696 #72-2712
ABOMSG	=	017120	97-3413 #97-3443
ABONM	=	006530	*49-1799 *49-1816 49-1821 #49-1829
ADR	=	000020 G	#10-640
ALLGON	=	003336 G	#15-857 *49-1825 *91-3278 95-3370
ASSEMB	=	000010	5-451 5-451
BDATA	=	000134 G	#17-940 89-3153 89-3163
BDBYTS	=	014572	*86-3001 *86-3012 86-3015 86-3020 #86-3030
BDCHK	=	000022 G	#11-668 66-2365 66-2422
BDCOM	=	000014 G	#11-665 68-2472 72-2684
BIT0	=	000001 G	#10-640 37-1615 60-2193 72-2675 78-2786 78-2787 78-2789 78-2800 82-2875
BIT00	=	000001 G	#10-640 10-640
BIT01	=	000002 G	#10-640 10-640
BIT02	=	000004 G	#10-640 10-640
BIT03	=	000010 G	#10-640 10-640 116-3795 116-3822
BIT04	=	000020 G	#10-640 10-640
BIT05	=	000040 G	#10-640 10-640
BIT06	=	000100 G	#10-640 10-640
BIT07	=	000200 G	#10-640 10-640
BIT08	=	000400 G	#10-640 10-640
BIT09	=	001000 G	#10-640 10-640
BIT1	=	000002 G	#10-640 49-1781 54-1972 60-2194 64-2294 68-2458 68-2529
BIT10	=	002000 G	#10-640 49-1779 50-1901 68-2478 68-2498 68-2506 68-2527 106-3596 106-3596
			106-3597 108-3639 108-3639 110-3670 112-3701 112-3701 114-3732
BIT11	=	004000 G	#10-640 89-3139 91-3215 91-3236 93-3337
BIT12	=	010000 G	#10-640 106-3596 106-3596 106-3596 108-3639 108-3639 108-3639 112-3701 112-3701
			112-3701
BIT13	=	020000 G	#10-640 49-1769 49-1777 66-2381 66-2393 68-2462
BIT14	=	040000 G	#10-640 49-1805 49-1808 70-2595 82-2925 91-3242
BIT15	=	100000 G	#10-640 37-1613 49-1788 56-2014 56-2036 58-2079 58-2098 58-2118 58-2142
			64-2302 68-2464 91-3214 91-3261 93-3337 97-3410
BIT2	=	000004 G	#10-640 49-1776 49-1790 54-1972 60-2195 64-2294 68-2458 68-2460 68-2529
BIT3	=	000010 G	#10-640 49-1767 60-2196 66-2345 66-2381 66-2393 102-3547 104-3565 106-3596
			106-3596 106-3596 106-3597 108-3639 108-3639 108-3639 110-3670 112-3701 112-3701
			112-3701 114-3732 116-3783 116-3812 116-3839
BIT4	=	000020 G	#10-640 50-1895 50-1899 60-2197 74-2731 82-2877
BIT5	=	000040 G	#10-640 50-1917 50-1919 60-2198
BIT6	=	000100 G	#10-640 60-2199 68-2469 68-2530 82-2902 82-2910 82-2917 82-2933 82-2934
			84-2946 84-2954 86-3021
BIT7	=	000200 G	#10-640 60-2200 68-2498 68-2506 68-2512 68-2514 68-2516 68-2527
BIT8	=	000400 G	#10-640 39-1648 49-1771 50-1905 50-1914 68-2491 68-2520 91-3243
BIT9	=	001000 G	#10-640 37-1617 50-1889 68-2475 68-2502 70-2585 91-3245
BLKEND	=	000202 G	#17-960 91-3253
BLKER	=	003330 G	#15-854 *50-1886 *91-3286
BLKSIZ	=	000212 G	#17-964 19-986 19-987 19-988 19-989 19-990 19-991 19-992 19-993
BLKTBL	=	003346 G	#19-974 37-1611 39-1645 41-1670 45-1713 49-1763 49-1800 58-2077 64-2300
			89-3133 91-3210 91-3230 93-3335 97-3434 116-3849
BOE	=	000400 G	#10-640
BRKPTR	=	014412	*82-2901 *82-2909 *82-2916 84-2967 #84-2981
BRKTO	=	014410	*82-2874 *82-2883 *82-2889 *82-2899 *82-2907 *84-2971 #84-2980
BRKWD	=	014404	82-2888 #84-2977

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
BUFTBL		026152	91-3217 #120-3888
BUFO		027232	120-3888 #120-3908
BUF1		030270	120-3889 #120-3909
BUF2		031326	120-3890 #120-3910
BUF3		032364	120-3891 #120-3911
BUF4		033422	120-3892 #120-3912
BUF5		034460	120-3893 #120-3913
BUF6		035516	120-3894 #120-3914
BUF7		036554	120-3895 #120-3915
CARLF		015010	88-3059 88-3061 #88-3070
CHECK		016140	91-3220 #91-3225
CHKANR		010546	64-2298 64-2310 #64-2314
CHKANS	G	010442	47-1738 #64-2292
CHKEND	G	011146	66-2370 66-2418 #68-2455
CHKERR		011252	68-2468 #68-2475
CHKPKS	G	010552	64-2296 64-2308 #66-2344
CHKPTR		010550	*64-2300 64-2301 64-2309 *64-2311 #64-2316
CHKREE		011324	68-2461 #68-2482
CHKRET		011670	68-2466 68-2471 68-2474 68-2477 68-2481 68-2499 68-2507 68-2515 68-2517
			#68-2529
CHKSUC	G	012164	68-2463 68-2482 #70-2564
CHKSUM	G	013520	#78-2784 80-2827 102-3547 104-3565 106-3596 106-3596 106-3596 106-3597 106-3597
			108-3639 108-3639 108-3639 110-3670 110-3670 112-3701 112-3701 112-3701 114-3732
			114-3732 116-3773 116-3801 116-3828
CHK8		010462	64-2295 #64-2300
CKCKSM	G	013614	66-2363 66-2401 66-2413 #80-2824
CLRALL	G	005706	#41-1670 54-1975
CLRBUF	G	005746	41-1672 #43-1688 54-1978
CLRPTR		006000	*41-1670 41-1671 41-1673 *41-1675 #43-1698
CMDSNT	=	000100	G #16-913 *50-1898 70-2571 70-2575 70-2588 74-2737 *82-2876
CMNDR	=	000040	G #11-675 70-2609
CMPDAT		002214	#9-566 86-3003
CNINIT	=	000032	G #11-672 82-2929
COMPAR		014414	G 66-2379 66-2403 #86-2998
CSNRDY		003342	G #15-864 52-1943
CSRCVB		003344	G #15-865 62-2230
C\$AU	=	000052	#5-451 99-3478
C\$AUTO	=	000061	#5-451 93-3346
C\$BRK	=	000022	#5-451 47-1736 49-1824 52-1947 62-2252 82-2879 84-2969 84-2970 89-3132
			89-3135 89-3137
C\$BSEG	=	000004	#5-451
C\$BSUB	=	000002	#5-451
C\$CEFG	=	000045	#5-451
C\$CLCK	=	000062	#5-451
C\$CLEA	=	000012	#5-451 95-3395
C\$CLOS	=	000035	#5-451
C\$CLP1	=	000006	#5-451
C\$CVEC	=	000036	#5-451 82-2935 82-2937 93-3345
C\$DCLN	=	000044	#5-451 49-1826 91-3228 91-3248
C\$DODU	=	000051	#5-451 72-2714 93-3357
C\$DRPT	=	000024	#5-451 95-3374
C\$DU	=	000053	#5-451 97-3433

SYMBOL	VALUE	CROSS REFERENCE	REFERENCES
C\$EDIT	= 000003		#5-451 5-494
C\$ERDF	= 000055		#5-451 72-2668 72-2682
C\$ERHR	= 000056		#5-451 72-2695 72-2710
C\$ERRO	= 000060		#5-451
C\$ERSF	= 000054		#5-451 49-1823 91-3227 91-3247
C\$ERSO	= 000057		#5-451 72-2688 72-2706 86-3019
C\$ESCA	= 000010		#5-451
C\$ESEG	= 000005		#5-451
C\$ESUB	= 000003		#5-451
C\$ETST	= 000001		#5-451 102-3552 104-3579 106-3621 108-3653 110-3684 112-3715 114-3746 116-3868
C\$EXIT	= 000032		#5-451 102-3545 104-3559 106-3589 108-3629 110-3660 112-3691 114-3722 116-3758
C\$GETB	= 000026		#5-451
C\$GETW	= 000027		#5-451
C\$GMAN	= 000043		#5-451
C\$GPHR	= 000042		#5-451 91-3234
C\$GPLO	= 000030		#5-451
C\$GPRI	= 000040		#5-451
C\$INIT	= 000011		#5-451 91-3315
C\$INLP	= 000020		#5-451
C\$MANI	= 000050		#5-451
C\$MEM	= 000031		#5-451
C\$MSG	= 000023		#5-451 74-2744
C\$OPEN	= 000034		#5-451
C\$PNTB	= 000014		#5-451 68-2504 74-2734 74-2737 74-2740 86-3020
C\$PNTF	= 000017		#5-451 88-3054 88-3059 88-3061 93-3354 97-3413 116-3848 116-3855 116-3857
			116-3859
C\$PNTS	= 000016		#5-451 89-3134 89-3136 89-3152 89-3153 89-3163
C\$PNTX	= 000015		#5-451 68-2480 68-2485 68-2488 68-2490 68-2511 68-2519
C\$QIO	= 000377		#5-451
C\$RDBU	= 000007		#5-451
C\$REFG	= 000047		#5-451 91-3207
C\$RESE	= 000033		#5-451 #5-451
C\$REVI	= 000003		#5-451 5-494
C\$RFLA	= 000021		#5-451 91-3285
C\$RPT	= 000025		#5-451 89-3176
C\$SEFG	= 000046		#5-451
C\$SPRI	= 000041		#5-451 82-2894
C\$SVEC	= 000037		#5-451 82-2895 82-2897 93-3334
C\$TPRI	= 000013		#5-451
DESC	014574		86-3020 #86-3031
DEVPTR	003310	G	#15-846 *49-1763 49-1764 49-1794 *49-1796 *49-1800 49-1801 49-1817 *49-1819
			*91-3210 91-3212 91-3219 *91-3221 *91-3230 91-3232 *91-3279 *116-3849 116-3850
			116-3860 *116-3862
DEV0	003366		19-974 #19-986
DEV1	003600		19-975 #19-987
DEV2	004012		19-976 #19-988
DEV3	004224		19-977 #19-989
DEV4	004436		19-978 #19-990
DEV5	004650		19-979 #19-991
DEV6	005062		19-980 #19-992
DEV7	005274		19-981 #19-993
DFPTBL	002174	G	#8-539

SYMBOL	VALUE		REFERENCES								
DFTL1	= 012640		*72-2680	*72-2681	#72-2682	72-2682					
DIAGMC	= 000000		5-451	5-451							
DLV	= 000074	G	#16-911	56-2034	58-2120	58-2144	*62-2233	62-2234	62-2235	*62-2237	*62-2258
			62-2259	62-2260	*62-2262	66-2383	74-2738	74-2740	*74-2741	*84-2955	84-2956
			84-2957	*84-2959	*91-3276						
DOBRK	013654	G	49-1787	49-1807	#82-2873	116-3781					
DONE	003320	G	*15-850	*45-1712	47-1732	*102-3548	*104-3570	*106-3612	*108-3650	*110-3681	*112-3712
			*114-3743	*116-3842							
DR	= 000060	G	#16-904	37-1615	37-1617	*37-1619	*39-1647	39-1648	*39-1650	74-2734	76-2765
			*91-3239	91-3243	91-3245	104-3565	106-3596	106-3596	106-3597	106-3597	108-3638
			108-3639	108-3639	110-3669	110-3670	110-3670	112-3700	112-3701	112-3701	114-3731
			114-3732	114-3732							
DRVCHK	002216		#9-567	108-3636	110-3667	112-3698	114-3729				
EF.COM	= 000036	G	#10-640								
EF.NEW	= 000035	G	#10-640								
EF.PWR	= 000034	G	#10-640								
EF.RES	= 000037	G	#10-640								
EF.STA	= 000040	G	#10-640	91-3207							
ENDGP8	010114		58-2076	58-2160	#58-2162						
ENDRSP	007076		50-1875	#50-1927							
ENDT8	025460		116-3846	116-3861	#116-3866						
ERRDES	013034	G	72-2668	72-2682	72-2688	72-2695	72-2706	72-2710	#74-2727	86-3019	
ESABO	= 177720	G	#12-734								
ESCKS	= 177757	G	#12-745	12-747	12-748	12-749	70-2583				
ESCKSM	= 177757		#12-747								
ESCPD	= 177720	G	#12-743	70-2607							
ESNCRT	= 177767	G	#12-735	70-2602							
ESNOMO	= 177737	G	#12-742	70-2578							
ESNONX	= 177770	G	#12-736	70-2612							
ESOK	= 000000	G	#12-737	70-2566							
ESPART	= 177776	G	#12-738	70-2622							
ESRD	= 177757		#12-749								
ESPEC	= 177711	G	#12-744	70-2627							
ES.	= 177740	G	#12-739	70-2596							
ESSLF	= 177777	G	#12-746								
ESTRY	= 000001	G	#12-740	70-2569							
ESWLOC	= 177765	G	#12-741	70-2617							
ESWR	= 177757		#12-748								
EVL	= 000004	G	#10-640	72-2676							
EVLTHR	002220		#9-568	72-2678							
EXOFF	007425		#56-2049	58-2151	62-2245						
EXON	007424		56-2005	#56-2048	58-2093	62-2253	62-2265				
ESEND	= 002100		#5-451								
ESLOAD	= 000035		#5-451	5-494							
FLGLOC	016616	G	72-2676	*91-3285	#91-3324						
FM	015574		89-3153	89-3163	#89-3186						
FMO	015556		89-3152	#89-3183							
FTLNM	003316		#15-849	72-2699							
FSAU	= 000015		#5-451	99-3455	99-3478						
FSAUTO	= 000020		#5-451	93-3332	93-3346						
FSBGN	= 000040		#5-451	5-477	6-503	9-580	10-633	74-2727	84-2944	84-2952	88-3073
			89-3117	89-3124	89-3192	91-3202	93-3332	95-3369	97-3405	99-3455	102-3540

CZTUUC SYMBOL	CROSS REFERENCE VALUE	REFERENCES	122-3998	122-3999	122-4000	122-4001	122-4002	124-4039	124-4040	124-4041
GSHILI	= 000002	#5-451								
GSLOLI	= 000001	#5-451								
GSNO	= 000000	#5-451								
GSOFFS	= 000400	#5-451	122-3998	122-3999	122-4000	122-4001	122-4002	124-4039	124-4040	124-4041
		124-4042	124-4043	124-4044	124-4045					
GSOF SI	= 000376	#5-451	122-3998	122-3999	122-4000	122-4001	122-4002	124-4039	124-4040	124-4041
		124-4042	124-4043	124-4044	124-4045					
GSPRMA	= 000001	#5-451	122-3998	122-3999						
GSPRMD	= 000002	#5-451	124-4039	124-4044						
GSPRML	= 000000	#5-451	122-4000	122-4001	122-4002	124-4040	124-4041	124-4042	124-4043	124-4045
GSRADA	= 000140	#5-451								
GSRADB	= 000000	#5-451								
GSRADD	= 000040	#5-451	124-4039	124-4044						
GSRADL	= 000120	#5-451	122-4000	122-4001	122-4002	124-4040	124-4041	124-4042	124-4043	124-4045
GSRADO	= 000020	#5-451	122-3998	122-3999						
GSXFER	= 000004	#5-451								
GSYES	= 000010	#5-451	122-3998	122-3999	122-4000	122-4001	122-4002	124-4039	124-4040	124-4041
		124-4042	124-4043	124-4044	124-4045					
HARDR	= 000136	G #17-941	89-3148	89-3158						
HARDW	= 000140	G #17-942	89-3150	89-3160						
HELP	= 000000	#5-436	5-446	5-468	5-486	7-509	7-524	8-546	9-571	#10-585
		10-623	10-642	19-994	19-1000	21-1016	21-1021	21-1029	21-1036	21-1041
		21-1047	37-1543	37-1555	37-1560	37-1566	37-1571	37-1577	37-1585	37-1592
		37-1598	37-1604	#89-3079	91-3293	91-3303	95-3376	95-3383	97-3415	97-3421
		99-3458	99-3464	#100-3484	100-3525	100-3531	120-3920	120-3925	120-3935	#122-3946
		122-4004	122-4020	124-4046	126-4066					
HOE	= 100000	G #10-640								
HRD	= 012772	72-2700	#72-2708							
HRDRD	= 000016	G #11-666	68-2522							
HRDWR	= 000020	G #11-667	68-2524							
HRD1	= 011612	68-2509	#68-2518							
IBE	= 010000	G #10-640								
IDPTR	= 003322	G #15-851	*45-1713	45-1714	45-1716	*45-1718				
IDU	= 000040	G #10-640								
IER	= 020000	G #10-640								
INIT	= 016030	#91-3204								
INITWD	= 014406	*82-2873	82-2900	82-2908	82-2915	82-2922	82-2928	#84-2978		
INIT2	= 016056	91-3208	#91-3210							
ISR	= 000100	G #10-640								
IXE	= 004000	G #10-640								
ISAU	= 000041	#5-451	#99-3455	#99-3478						
ISAUTO	= 000041	#5-451	#93-3332	#93-3346						
ISCLN	= 000041	#5-451	#95-3369	#95-3395						
ISDU	= 000041	#5-451	#97-3405	#97-3433						
ISHRD	= 000041	#122-3995	#122-4010							
ISINIT	= 000041	#5-451	#91-3202	#91-3315						
ISMOD	= 000041	#5-451	5-477	#5-477	9-580	#9-580	10-633	#10-633	88-3073	#88-3073
		89-3117	#89-3117	89-3192	#89-3192	102-3540	#102-3540	120-3919	#120-3919	122-3984
		#122-3984	126-4073	#126-4073						
ISMSG	= 000041	#5-451	#74-2727	#74-2744						
ISPROT	= 000040	#5-451	#6-503							
ISPTAB	= 000041	#5-451	126-4076	#126-4076	126-4081	#126-4081				

CZTUUC SYMBOL	CREATED BY	MACRO	ON 25-SEP-81 AT 10:08	PAGE 7	M 10						SEQ 0129
SYMBOL	CROSS REFERENCE	VALUE	REFERENCES	CREF							
ISPWR	=	000041	#5-451								
ISRPT	=	000041	#5-451	#89-3124	#89-3176						
ISSEG	=	000041	#5-451	102-3543	104-3557	106-3587	108-3627	110-3658	112-3689	114-3720	116-3751
ISSETU	=	000041	#5-451	126-4075	#126-4075	126-4076	126-4082	#126-4082			
ISSFT	=	000041	#124-4038	#124-4052							
ISSRV	=	000041	#5-451	#84-2944	#84-2948	#84-2952	#84-2961				
ISSUB	=	000041	#5-451	102-3543	104-3557	106-3587	108-3627	110-3658	112-3689	114-3720	116-3751
ISTST	=	000041	#5-451	102-3543	#102-3543	102-3545	102-3552	#102-3552	#102-3552	104-3557	#104-3557
				104-3559	104-3579	#104-3579	#104-3579	106-3587	#106-3587	106-3589	106-3621
				#106-3621	108-3627	#108-3627	108-3629	108-3653	#108-3653	#108-3653	110-3658
				110-3660	110-3684	#110-3684	#110-3684	112-3689	#112-3689	112-3691	112-3715
				#112-3715	114-3720	#114-3720	114-3722	114-3746	#114-3746	#114-3746	116-3751
				116-3758	116-3868	#116-3868	#116-3868				
JSJMP	=	000167	#5-451								
LENGTH	=	002206	#9-563	91-3287							
LGOFST	=	000120	G	#17-931	72-2660						
LNCNT	=	014774	G	*88-3052	*88-3057	#88-3066					
LOE	=	040000	G	#10-640							
LOG	=	012500	G	52-1953	62-2273	66-2366	66-2386	66-2395	66-2423	68-2473	68-2496
				70-2633	#72-2652	82-2886	82-2930	84-2974			68-2525
LOGO	=	013022		72-2689	72-2707	#72-2715					
LOGOK	=	012566		72-2667	#72-2670						
LOGOK2	=	012652		72-2677	72-2679	#72-2684					
LOGO	=	012554		#72-2668							
LOG1	=	012670		*72-2686	*72-2687	#72-2688	72-2688				
LOG2	=	012720		*72-2693	*72-2694	#72-2695	72-2695				
LOG3	=	012760		*72-2704	*72-2705	#72-2706	72-2706				
LOG3B	=	013002		*72-2708	*72-2709	#72-2710	72-2710				
LOT	=	000010	G	#10-640							
LSTDEV	=	003364	G	#19-981	37-1621	39-1651	41-1673	45-1716	49-1794	49-1817	58-2154
				89-3164	91-3219	93-3341	116-3860				64-2309
LSACP	=	002110	G	#5-494							
LSAPT	=	002036	G	#5-494							
LSAU	=	017146	G	5-494	#99-3455						
LSAUT	=	002070	G	#5-494							
LSAUTO	=	016620	G	5-494	#93-3332						
LSCCP	=	002106	G	#5-494							
LSCLEA	=	017002	G	5-494	#95-3369						
LSCO	=	002032	G	#5-494							
LSDEPO	=	002011	G	#5-494							
LSDESC	=	002122	G	5-494	#5-496						
LSDESP	=	002076	G	#5-494							
LSDEVP	=	002060	G	#5-494							
LSDISP	=	002152	G	5-494	#7-522						
LSDLY	=	002116	G	#5-494							
LSDTP	=	002040	G	#5-494							
LSDTYP	=	002034	G	#5-494							
LSDU	=	017022	G	5-494	#97-3405						
LSDUT	=	002072	G	#5-494							
LSDVTY	=	005506	G	5-494	#21-1014						
LSEF	=	002052	G	#5-494							
LSENV1	=	002044	G	#5-494							

CZTUUC
SYMBOL
SYMBOL

CREATED BY MACRO ON 25-SEP-81 AT 10:08

PAGE 8
CREF

N 10

SEQ 0130

SYMBOL	VALUE		REFERENCES
L\$ETP	002102	G	#5-494
L\$EXP1	002046	G	#5-494
L\$EXP4	002064	G	#5-494
L\$EXP5	002066	G	#5-494
L\$HARD	037614	G	5-494 122-3995 #122-3995
L\$HIME	002120	G	#5-494
L\$HPCP	002016	G	#5-494
L\$HPTP	002022	G	#5-494
L\$HW	002174	G	5-494 8-539 #8-539
L\$ICP	002104	G	#5-494
L\$INIT	016030	G	5-494 #91-3202
L\$LDAP	002026	G	#5-494
L\$LAST	040422	G	5-494 #126-4072 126-4082
L\$LOAD	002100	G	#5-494
L\$LUN	002074	G	#5-494 *72-2657 *72-2658 *86-3017 *86-3018 *91-3233
L\$MREV	002050	G	#5-494
L\$NAME	002000	G	#5-494
L\$PRIO	002042	G	#5-494
L\$PROT	002142	G	5-494 #6-503
L\$PRT	002112	G	#5-494
L\$REPP	002062	G	#5-494
L\$REV	002010	G	#5-494
L\$RPT	015014	G	5-494 #89-3124
L\$SOFT	037756	G	5-494 124-4038 #124-4038
L\$SPC	002056	G	#5-494
L\$SPCP	002020	G	#5-494
L\$SPTP	002024	G	#5-494
L\$STA	002030	G	#5-494
L\$SW	002206	G	5-494 9-561 #9-561
L\$TEST	002114	G	#5-494
L\$TIML	002014	G	#5-494
L\$UNIT	002012	G	#5-494 91-3225 91-3281
L10001	002204		8-539 #8-552
L10002	002224		9-561 #9-578
L10003	013224		#74-2744
L10004	014312		#84-2948
L10005	014346		#84-2961
L10006	015436		#89-3176
L10007	016526		#91-3315
L10010	016724		#93-3346
L10011	017020		#95-3395
L10012	017064		#97-3433
L10013	017146		#99-3478
L10014	017350		102-3545 #102-3552
L10015	017622		104-3559 #104-3579
L10016	021226		106-3589 #106-3621
L10017	022216		108-3629 #108-3653
L10020	023002		110-3660 #110-3684
L10021	023772		112-3691 #112-3715
L10022	024556		114-3722 #114-3746
L10023	025462		116-3758 #116-3868
L10024	037656		122-3995 #122-4010

C
E
R
T
I
F
I
C
A
T
E
D

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES						
L10025		040040	124-4038	#124-4052					
L10026		040426	#126-4076						
L10030		040436	126-4076	#126-4081					
MABEE		012732	72-2692	#72-2698					
MODRSP		010116	58-2129	#58-2164					
MRSPLY		010120	*58-2132	*58-2136	#58-2166				
MRSP	=	000210	#17-963	50-1874	58-2074	64-2306	*66-2377	*66-2417	*91-3277
MSAGE1		025464	116-3848	#116-3870					116-3853
MSAGE2		025525	116-3855	#116-3871					
MSAGE3		025561	116-3857	#116-3872					
MSAGE4		025610	116-3859	#116-3873					
MSAUTO		016762	93-3354	#93-3359					
MSBDA		002336	#14-788	86-3019					
MSCMD		002702	13-773	#14-810					
MSCOM		002402	13-763	#14-792					
MSG1		037656	122-3998	#122-4012					
MSG1B		037667	122-3999	#122-4013					
MSG1C		037704	122-4000	#122-4014					
MSG2		037722	122-4001	#122-4015					
MSG3		037737	122-4002	#122-4016					
MSG4		040040	124-4039	#124-4053					
MSG4B		040105	124-4040	#124-4054					
MSG5		040147	124-4041	#124-4055					
MSG6		040201	124-4042	#124-4056					
MSG7		040226	124-4043	#124-4057					
MSG8		040254	124-4044	#124-4058					
MSG9		040315	124-4045	#124-4059					
MSHCHK		002554	13-766	#14-802					
MSHDRD		003152	13-764	#14-826					
MSHDWR		003214	13-765	#14-828					
MSNIT		002616	13-770	#14-804					
MSNLOG		002320	13-757	13-778	#14-786				
MSNOMO		002444	13-769	#14-794					
MSNOTP		002462	13-779	#14-796					
MSNRSP		002762	13-777	#14-816					
MSOVRN		003256	13-762	#14-830					
MSPART		002632	13-771	#14-806					
MSQRSP		002776	13-761	#14-818					
MSREC		002716	13-774	#14-812					
MSRNIT		002534	13-760	#14-800					
MSELF		002362	13-775	#14-790					
MSSFRD		003052	13-758	#14-822					
MSSFWR		003112	13-759	#14-824					
MSSKER		002304	13-767	#14-784					
MSTOSN		003030	13-780	#14-820					
MSUNIT		002654	13-772	#14-808					
MSWPRO		002512	13-768	#14-798					
MSWRSP		002736	13-776	#14-814					
MXRTRY		003326	#15-853	68-2508					
NCART	=	000054	#11-680	70-2604					
NODRVS		016560	91-3247	#91-3320					
NOMOR		006532	49-1823	#49-1830					

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
NOMOT	= 000030	G	#11-671 70-2580
NOREE	011200		68-2459 #68-2463
NOUNIT	= 000036	G	#11-674 70-2614
NOXOFF	006630		#50-1876
NTSFT	012702		72-2685 #72-2691
NXTRET	006526		49-1793 49-1822 #49-1827
NXTST	006100	G	47-1730 #49-1762
NXTST2	006350		49-1795 #49-1799
ONEFIL	= 000001		#2-4 2-8 4-432 5-433 5-472 9-581 10-582 10-595 88-3075 89-3076 89-3089 99-3480 100-3481 100-3492 121-3941 122-3942 122-3956
OTL	= 000010	G	#11-663 66-2382
OVRFLO	013400		72-2668 #74-2749
OVRN	= 000012	G	#11-664 66-2385
OSAPTS	= 000000		#5-451 5-494
OSAU	= 000001		#5-451 #5-484 5-494
OSBGNR	= 000001		#5-451 #5-484 5-494
OSBGNS	= 000001		#5-451 #5-484 5-494
OSDU	= 000001		#5-451 #5-484 5-494
OSERRT	= 000000		#5-451 5-494
OSGNSW	= 000001		#5-451 #5-484 5-494
OSPOIN	= 000001		#5-451 #5-484 #5-484 #5-484 #5-484 #5-484 5-484 5-494
OSSETU	= 000001		#5-451 #5-484 5-494 126-4072
PARTL	= 000034	G	#11-673 70-2624
PATTEN	= 000072	G	#16-910 74-2736 86-3010 *91-3272 *106-3595 106-3596 *108-3635 *108-3638 108-3639 *110-3666 *110-3669 *112-3697 *112-3700 112-3701 *114-3728 *114-3731
PDTFLG	016614	G	*91-3241 91-3268 #91-3323
PERDEV	006362		#49-1801 49-1820
PKPTR	= 000104	G	#16-916 *50-1890 58-2096 58-2148 *58-2150 *66-2352 86-3002 88-3051
PNT	= 001000	G	#10-640
PPSOT8	002222		#9-569 116-3845
PRBUF	002212		#9-565 88-3049
PRDAT	014776		*88-3053 88-3054 #88-3067
PRFORM	015000		88-3054 #88-3068
PRI	= 002000	G	#10-640
PRI00	= 000000	G	#10-640 82-2894
PRI01	= 000040	G	#10-640
PRI02	= 000100	G	#10-640
PRI03	= 000140	G	#10-640
PRI04	= 000200	G	#10-640
PRI05	= 000240	G	#10-640
PRI06	= 000300	G	#10-640
PRI07	= 000340	G	5-494 #10-640 82-2895 82-2897 93-3334
PRNPAK	014630	G	86-3023 #88-3044
PRNSIZ	003334	G	#15-856 *86-3022 *88-3055
PTR	017116		*97-3434 97-3435 *97-3438 #97-3441
RCBCNT	003314		#15-848 *56-2011 *56-2022 *56-2027 *56-2031 *58-2090 *58-2109 *58-2113 *58-2115 *58-2139 *66-2354 66-2361 66-2428
RCBFSZ	= 001036	G	#12-718 43-1691 120-3904 120-3908 120-3909 120-3910 120-3911 120-3912 120-3913 120-3914 120-3915
RCDB	= 000024	G	#16-896 62-2233 62-2258 82-2893 84-2955 *91-3264
RCFLG	003312	G	#15-847 *56-2012 56-2018 *58-2088 58-2104 *66-2355
RCINIT	= 000006	G	#11-662 66-2394

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
RCSR	= 000022	G	#16-895 62-2231 62-2247 82-2917 82-2934 84-2954 *91-3262 93-3339
RCVBUF	= 000102	G	#16-915 43-1690 50-1890 56-2008 66-2346 *91-3217
RCVHND	014314		#84-2954
RCVINT	014314	G	82-2895 #84-2952
RDNO	= 000114	G	#16-920 *50-1908 89-3153
RDNI	= 000116	G	#16-921 *50-1910 89-3163
REC	= 000064	G	#16-906 50-1886 74-2735 *91-3274 *104-3563 104-3565 *106-3594 106-3596 106-3596 106-3597 106-3597 *106-3603 106-3604 *106-3607 *106-3608 *108-3532 108-3635 108-3639 108-3639 *108-3642 *108-3647 *110-3663 110-3666 110-3670 110-3670 *110-3673 *110-3678 *112-3694 112-3697 112-3701 112-3701 *112-3704 *112-3709 *114-3725 114-3728 114-3732 114-3732 *114-3735 *114-3740 116-3792 116-3819 104-3563 104-3568 #104-3573
RECDAT	017606		
RECERR	= 000042	G	#11-676 70-2629
RECID	013306	G	74-2737 #74-2747
RECID2	013462		74-2740 #74-2751
RECOV	011710		68-2485 #68-2536
RETErr	012112		68-2504 #68-2546
RETRY	= 000002	G	#16-887 *68-2479 68-2480 68-2485 *68-2497 *68-2505 68-2508 *68-2510 68-2511 *68-2526 *91-3273
RLUN	015440		*89-3142 *89-3143 89-3152 #89-3177
RPTR	015442		*89-3133 89-3138 89-3164 *89-3166 #89-3178
RSCMND	= 000002	G	#12-698 12-704 50-1896 102-3547 104-3565 104-3565 106-3596 106-3596 106-3597 106-3597 108-3639 108-3639 110-3670 110-3670 112-3701 112-3701 114-3732 114-3732 116-3762 116-3789 116-3805 116-3816 116-3832
RSCONT	= 000020	G	#12-699 58-2164 66-2358 82-2922 106-3596 106-3596 108-3639 108-3639 112-3701
RSDASZ	= 000204	G	#12-711 12-713 12-718 56-2027 66-2399 66-2406
RSDATA	= 000001	G	#12-703 56-2025 58-2111 58-2148 66-2373 66-2397 106-3596 106-3596 106-3597 106-3597 108-3639 108-3639 110-3670 110-3670 112-3701 112-3701 114-3732 114-3732 116-3775
RSDNSZ	= 000222	G	#12-713 58-2113
RSEND	= 000002	G	#12-704 56-2020 58-2107 66-2368 66-2409 102-3547 106-3596 106-3596 106-3597 106-3597 108-3639 108-3639 110-3670 110-3670 112-3701 112-3701 114-3732 114-3732
RSGCDP	= 000034	G	#12-715 116-3776
RSINIT	= 000004	G	#12-702 66-2391 84-2978
RSMSIZ	= 000012	G	#12-709 12-717 102-3547 102-3547 104-3565 104-3565 106-3596 106-3596 106-3596 106-3596 106-3597 106-3597 106-3597 108-3639 108-3639 108-3639 108-3639 108-3639 110-3670 110-3670 110-3670 110-3670 112-3701 112-3701 112-3701 112-3701 114-3732 114-3732 114-3732 114-3732 116-3763 116-3770 116-3790 116-3798 116-3817 116-3825
RSNDSZ	= 000016	G	#12-707 12-713 12-718 56-2022 58-2109 102-3547 104-3565 106-3596 106-3596 106-3597 106-3597 108-3639 108-3639 110-3670 110-3670 112-3701 112-3701 114-3732 114-3732 116-3806 116-3833
RSNTAB	002224		#13-757 72-2673
RSEND	= 000100	G	#12-723
RSSGET	:: 000012	G	#12-727 116-3764
RSSNIT	= 000001	G	#12-729 82-2876 116-3791
RSSNOP	= 000000	G	#12-728 116-3818
RSSNSZ	= 000016	G	#12-717 66-2411 102-3547 104-3565 106-3596 106-3596 106-3596 106-3597 106-3597 108-3639 108-3639 110-3670 110-3670 112-3701 112-3701 114-3732 114-3732 116-3772 116-3804 116-3831
RSSRD	= 000002	G	#12-725 50-1903 70-2571 106-3596 106-3597 106-3597 108-3639 110-3670 110-3670 112-3701 114-3732 114-3732

REFERENCES

SYMBOL	VALUE	REFERENCES
		106-3621 106-3621 108-3629 108-3629 108-3629 108-3629 108-3629 108-3629 108-3629 108-3653
		108-3653 108-3653 110-3660 110-3660 110-3660 110-3660 110-3660 110-3660 110-3660 110-3684
		110-3684 110-3684 112-3691 112-3691 112-3691 112-3691 112-3691 112-3691 112-3691 112-3715
		112-3715 112-3715 114-3722 114-3722 114-3722 114-3722 114-3722 114-3722 114-3722 114-3746
		114-3746 114-3746 116-3758 116-3758 116-3758 116-3758 116-3758 116-3758 116-3758 116-3848
		116-3848 116-3848 116-3848 116-3848 116-3848 116-3848 116-3848 116-3848 116-3848 116-3848
		116-3848 116-3848 116-3848 116-3848 116-3848 116-3848 116-3848 116-3848 116-3848 116-3848
		116-3855 116-3855 116-3855 116-3855 116-3855 116-3855 116-3855 116-3855 116-3855 116-3855
		116-3855 116-3855 116-3855 116-3855 116-3855 116-3855 116-3855 116-3855 116-3855 116-3855
		116-3857 116-3857 116-3857 116-3857 116-3857 116-3857 116-3857 116-3857 116-3857 116-3857
		116-3857 116-3857 116-3857 116-3857 116-3857 116-3857 116-3857 116-3857 116-3857 116-3857
		116-3859 116-3859 116-3859 116-3859 116-3859 116-3859 116-3859 116-3859 116-3859 116-3859
		116-3859 116-3859 116-3859 116-3859 116-3859 116-3859 116-3859 116-3859 116-3859 116-3859
		122-3995 122-3995 122-3998 122-3998 122-3998 122-3998 122-3998 122-3998 122-3998 122-3995
		122-3998 122-3998 122-3998 122-3998 122-3998 122-3998 122-3998 122-3998 122-3998 122-3998
		122-3999 122-3999 122-3999 122-3999 122-3999 122-3999 122-3999 122-3999 122-3999 122-3999
		122-4000 122-4000 122-4000 122-4000 122-4000 122-4000 122-4000 122-4000 122-4000 122-4000
		122-4001 122-4001 122-4001 122-4001 122-4001 122-4001 122-4001 122-4001 122-4001 122-4001
		122-4002 122-4002 122-4002 122-4002 122-4002 122-4002 122-4002 122-4002 122-4002 122-4002
		122-4010 122-4010 124-4038 124-4038 124-4038 124-4038 124-4038 124-4038 124-4038 124-4039
		124-4039 124-4039 124-4039 124-4039 124-4039 124-4039 124-4039 124-4039 124-4039 124-4039
		124-4039 124-4039 124-4039 124-4039 124-4039 124-4039 124-4039 124-4039 124-4039 124-4039
		124-4040 124-4040 124-4041 124-4041 124-4041 124-4041 124-4041 124-4041 124-4041 124-4041
		124-4041 124-4041 124-4042 124-4042 124-4042 124-4042 124-4042 124-4042 124-4042 124-4042
		124-4042 124-4042 124-4043 124-4043 124-4043 124-4043 124-4043 124-4043 124-4043 124-4043
		124-4043 124-4043 124-4044 124-4044 124-4044 124-4044 124-4044 124-4044 124-4044 124-4044
		124-4044 124-4044 124-4044 124-4044 124-4044 124-4044 124-4044 124-4044 124-4044 124-4044
		124-4045 124-4045 124-4045 124-4045 124-4045 124-4045 124-4045 124-4045 124-4045 124-4045
		124-4052 124-4052 126-4072 126-4072 126-4072 126-4072 126-4072 126-4072 126-4072 126-4072
		126-4072 126-4072 126-4076 126-4076 126-4076 126-4076 126-4076 126-4076 126-4076 126-4076
SVCSUB	= 000001	#5-451 #5-459
SVCTAG	= 000001	#5-451 #5-461 8-552 9-578 74-2744 84-2948 84-2961 89-3176 91-3315
		93-3346 95-3395 97-3433 99-3478 102-3552 104-3579 106-3621 108-3653 110-3684
		112-3715 114-3746 116-3868 122-4010 124-4052 126-4076 126-4081 108-3653 110-3684
SVCTST	= 000001	#5-451 #5-458 102-3543 104-3557 106-3587 108-3627 110-3658 112-3689 114-3720
		116-3751
SWAPDR	005526 G	#37-1610 102-3544 104-3558 106-3588 108-3628 110-3659 112-3690 114-3721
SWPTR	005626	*37-1611 37-1612 37-1621 *37-1623 #37-1631
SYSTAT	003304 G	#15-836 *49-1776 *49-1781 *49-1790 54-1972 *56-2017 *58-2103 64-2294 *66-2353
		68-2458 68-2460 *68-2529 74-2734 *78-2786 *78-2789 78-2800 *82-2928 *91-3284
SLSYM	= 010000	#5-451 #8-552 #9-578 #74-2744 #84-2948 #84-2961 #89-3176 #91-3315 #93-3346
		#95-3395 #97-3433 #99-3478 #102-3552 #104-3579 #106-3621 #108-3653 #110-3684 #112-3715
		#114-3746 #116-3868 #122-4010 #124-4052
TAPLEN	003306 G	#15-845 *91-3287 *91-3288 91-3290 108-3633 108-3648 110-3664 110-3679 112-3695
		112-3710 114-3726 114-3741
TEST8	003340 G	#15-858 50-1872 58-2072 58-2091 58-2126 64-2304 66-2375 66-2415 *91-3206
		*116-3760 *116-3786 *116-3843
THRSHI	012016	68-2490 #68-2540
THRSLO	011770	68-2488 #68-2538
TMP	= 000066 G	#16-908 *106-3592 106-3608 *106-3609 *108-3633 *108-3640 *108-3648 *110-3664 *110-3671
		*110-3679 *112-3695 *112-3702 *112-3710 *114-3726 *114-3733 *114-3741
TOMANY	016530	91-3227 #91-3318

CZTUUC SYMBOL	CROSS REFERENCE VALUE	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES
TORCVB	= 000050 G	#11-679	62-2272	84-2973						
TOSNDB	= 000056 G	#11-681	52-1952	82-2885						
TRBUF	026174	50-1876	50-1878	50-1885	102-3547	104-3565	106-3596	106-3596	106-3596	106-3596
		106-3597	106-3597	108-3639	108-3639	108-3639	108-3639	110-3670	110-3670	112-3701
		112-3701	112-3701	112-3701	114-3732	114-3732	116-3761	116-3788	116-3815	#120-3904
TRK	= 000062 G	#16-905	*108-3634	108-3644	*108-3646	*110-3665	110-3675	*110-3677	*112-3696	112-3706
		*112-3708	*114-3727	114-3737	*114-3739					
TRPHND	016730	93-3334	#93-3354							
TRPPTR	016726	*93-3335	93-3336	93-3341	*93-3343	#93-3347				
TSTPC	= 000020 G	#16-894	*45-1715	49-1792	49-1813	*50-1868				
TSTTOP	003324	#15-852	45-1715	*102-3544	*104-3558	*106-3588	*108-3628	*110-3659	*112-3690	*114-3721
		*116-3753								
TST1	017214	102-3544	#102-3547							
TST2	017416	104-3558	#104-3562							
TST3	017670	106-3588	#106-3592							
TST3PT	021214	106-3595	106-3599	#106-3614						
TST4	021274	108-3628	#108-3632							
TST4EX	022210	108-3645	#108-3650							
TST5	022264	110-3659	#110-3663							
TST5EX	022774	110-3676	#110-3681							
TST6	023050	112-3690	#112-3694							
TST6EX	023764	112-3707	#112-3712							
TST7	024040	114-3721	#114-3725							
TST7EX	024550	114-3738	#114-3743							
TST8	024602	116-3753	#116-3760							
TUVECT	= 000204 G	#17-961	82-2895	*82-2896	82-2897	*82-2898	82-2935	*82-2936	82-2937	*82-2938
		*91-3238								
T\$ARGC	= 000002	#5-494	5-494	#5-494	5-494	5-494	#5-494	5-494	5-494	#5-494
		5-494	5-494	#5-494	5-494	5-494	#5-494	5-494	5-494	#68-2480
		68-2480	#68-2480	68-2480	68-2480	#68-2485	68-2485	#68-2485	68-2485	68-2485
		#68-2488	68-2488	68-2488	#68-2490	68-2490	68-2490	#68-2504	68-2504	68-2504
		#68-2511	68-2511	#68-2511	68-2511	68-2511	#68-2519	68-2519	68-2519	#74-2734
		74-2734	#74-2734	74-2734	#74-2734	74-2734	#74-2734	74-2734	74-2734	#74-2737
		71-2737	#74-2737	74-2737	#74-2737	74-2737	#74-2737	74-2737	#74-2737	74-2737
		74-2737	#74-2740	74-2740	#74-2740	74-2740	74-2740	#86-3020	86-3020	#86-3020
		86-3020	86-3020	#88-3054	88-3054	#88-3054	88-3054	88-3054	#88-3059	88-3059
		88-3059	#88-3061	88-3061	88-3061	#89-3134	89-3134	89-3134	#89-3136	89-3136
		89-3136	#89-3152	89-3152	#89-3152	89-3152	89-3152	#89-3153	89-3153	#89-3153
		89-3153	#89-3153	89-3153	#89-3153	89-3153	#89-3153	89-3153	#89-3153	89-3153
		#89-3153	89-3153	#89-3153	89-3153	#89-3153	89-3153	#89-3153	89-3153	#89-3163
		#89-3163	89-3163	#89-3163	89-3163	#89-3163	89-3163	#89-3163	89-3163	#89-3163
		89-3163	#89-3163	89-3163	#89-3163	89-3163	#89-3163	89-3163	89-3163	#93-3354
		93-3354	93-3354	#97-3413	97-3413	#97-3413	97-3413	97-3413	#116-3848	116-3848
		116-3848	#116-3855	116-3855	#116-3855	116-3855	116-3855	#116-3857	116-3857	#116-3857
		116-3857	116-3857	#116-3859	116-3859	#116-3859	116-3859	116-3859	116-3859	116-3859
T\$CODE	= 006130	#122-3998	122-3998	#122-3998	122-3998	#122-3998	122-3998	#122-3999	122-3999	#122-3999
		122-3999	#122-3999	122-3999	#122-4000	122-4000	#122-4000	122-4000	#122-4000	122-4000
		#122-4001	122-4001	#122-4001	122-4001	#122-4001	122-4001	#122-4002	122-4002	#122-4002
		122-4002	#122-4002	122-4002	#124-4039	124-4039	#124-4039	124-4039	#124-4039	124-4039
		#124-4040	124-4040	#124-4040	124-4040	#124-4040	124-4040	#124-4041	124-4041	#124-4041
		124-4041	#124-4041	124-4041	#124-4042	124-4042	#124-4042	124-4042	#124-4042	124-4042
		#124-4043	124-4043	#124-4043	124-4043	#124-4043	124-4043	#124-4044	124-4044	#124-4044

CZTUUC SYMBOL CROSS REFERENCE SYMBOL VALUE

CREATED BY MACRO ON 25-SEP-81 AT 10:08

PAGE 17 CREF

SEQ 0139

SYMBOL	VALUE	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES	REFERENCES
TSERRN	= 000146	124-4044	#124-4044	124-4044	#124-4045	124-4045	#124-4045	124-4045	#124-4045
		#5-451	#49-1823	49-1823	#72-2668	72-2668	#72-2682	72-2682	#72-2688
		#72-2695	72-2695	#72-2706	72-2706	#72-2710	72-2710	#86-3019	86-3019
		91-3227	#91-3247	91-3247					#91-3227
TSEXCP	= 000000	#122-3998	122-3998	#122-3999	122-3999	#124-4039	124-4039	#124-4044	124-4044
TSFLAG	= 000040	#102-3545	#102-3545	102-3545	102-3545	#104-3559	#104-3559	104-3559	104-3559
		#106-3589	106-3589	106-3589	#108-3629	#108-3629	108-3629	108-3629	#110-3660
		110-3660	110-3660	#112-3691	#112-3691	112-3691	112-3691	#114-3722	#114-3722
		114-3722	#116-3758	#116-3758	116-3758	116-3758	116-3758		114-3722
TSFREE	= 040436	126-4072	#126-4082						
TSGMAN	= 000000	#5-451							
TSHILI	= 000376	#122-3998	122-3998	#122-3999	122-3999	#124-4039	124-4039	#124-4044	124-4044
TSLAST	= 000001	#5-451	#126-4072	126-4075					
TSLOLI	= 000001	#122-3998	122-3998	#122-3999	122-3999	#124-4039	124-4039	#124-4044	124-4044
TSLSYM	= 010000	#5-451	5-451	8-552	9-578	74-2744	84-2948	84-2961	89-3176
		93-3346	95-3395	97-3433	99-3478	102-3552	104-3579	106-3621	108-3653
		112-3715	114-3746	116-3868	122-4010	124-4052			91-3315
TSLTNO	= 000010	#126-4072							110-3684
TSNEST	= 177777	#5-451	5-477	#5-477	5-477	6-503	#6-503	6-503	6-507
		6-507	#6-507	8-539	#8-539	8-539	8-552	8-552	8-552
		9-561	#9-561	9-561	9-578	9-578	#9-578	#9-578	9-580
		9-580	#9-580	10-633	#10-633	10-633	74-2727	#74-2727	74-2727
		74-2744	74-2744	#74-2744	84-2944	#84-2944	84-2944	84-2948	84-2948
		#84-2948	84-2952	#84-2952	84-2952	84-2961	84-2961	84-2961	#84-2961
		88-3073	88-3073	#88-3073	89-3117	#89-3117	89-3117	89-3124	#89-3124
		89-3176	89-3176	89-3176	#89-3176	89-3192	89-3192	89-3192	#89-3192
		#91-3202	91-3202	91-3315	91-3315	91-3315	#91-3315	93-3332	#93-3332
		93-3346	93-3346	93-3346	#93-3346	95-3369	#95-3369	95-3369	95-3395
		95-3395	#95-3395	97-3405	#97-3405	97-3405	97-3433	97-3433	97-3433
		99-3455	#99-3455	99-3455	99-3478	99-3478	#99-3478	102-3540	#102-3540
		102-3540	102-3543	#102-3543	102-3543	102-3552	102-3552	102-3552	#102-3552
		#104-3557	104-3557	104-3579	104-3579	104-3579	#104-3579	106-3587	#106-3587
		106-3621	106-3621	106-3621	#106-3621	108-3627	#108-3627	108-3627	108-3653
		108-3653	#108-3653	110-3658	#110-3658	110-3658	110-3684	110-3684	110-3684
		112-3689	#112-3689	112-3689	112-3715	112-3715	112-3715	#112-3715	114-3720
		114-3720	114-3746	114-3746	114-3746	#114-3746	116-3751	#116-3751	116-3751
		116-3868	116-3868	#116-3868	120-3919	120-3919	120-3919	#120-3919	122-3984
		122-3984	122-3995	#122-3995	122-3995	122-4010	122-4010	122-4010	#122-4010
		#124-4038	124-4038	124-4052	124-4052	124-4052	#124-4052	126-4073	126-4073
		#126-4073							
TSNSO	= 000000	#5-477	9-580	#10-633	88-3073	#89-3117	89-3192	#91-3202	91-3315
		93-3346	#95-3369	95-3395	#97-3405	97-3433	#99-3455	99-3478	#102-3540
		#122-3984	126-4073						120-3919
TSNS1	= 000005	#6-503	6-507	#8-539	8-552	#9-561	9-578	#74-2727	74-2744
		84-2948	#84-2952	84-2961	#89-3124	89-3176	#102-3543	102-3552	#104-3557
		#106-3587	106-3621	#108-3627	108-3653	#110-3658	110-3684	#112-3689	112-3715
		114-3746	#116-3751	116-3868	#122-3995	122-4010	#124-4038	124-4052	#114-3720
		#126-4075	126-4076	#126-4076	126-4076				
TSPCNT	= 000000	#126-4076	126-4076						
TSPTAB	= 010027	5-494	#126-4082						
TSPTHV	= 000001	#5-451	126-4076	#126-4076	126-4082	126-4082			
TSPTNU	= 000001	#5-451							
TSSAVL	= 177777	#5-451							

CZTUUC
 SYMBOL CROSS REFERENCE
 SYMBOL VALUE
 T\$SEGL = 177777
 T\$SIZE = 000006
 T\$SUBN = 000000
 T\$TAGL = 177777
 T\$TAGN = 010031

REFERENCES
 #5-451
 126-4072 #126-4082
 #5-451 #102-3543 #104-3557 #106-3587 #108-3627 #110-3658 #112-3689 #114-3720 #116-3751
 #5-451
 #5-451 6-503 6-503 #6-503 8-539 8-539 #8-539 9-561 9-561
 #9-561 74-2727 74-2727 #74-2727 84-2944 84-2944 #84-2944 84-2952 84-2952
 #84-2952 89-3124 89-3124 #89-3124 91-3202 91-3202 #91-3202 93-3332 93-3332
 #93-3332 95-3369 95-3369 #95-3369 97-3405 97-3405 #97-3405 99-3455 99-3455
 #99-3455 102-3543 102-3543 #102-3543 104-3557 104-3557 #104-3557 106-3587 106-3587
 #106-3587 108-3627 108-3627 #108-3627 110-3658 110-3658 #110-3658 112-3689 112-3689
 #112-3689 114-3720 114-3720 #114-3720 116-3751 116-3751 #116-3751 122-3995 122-3995
 #122-3995 124-4038 124-4038 #124-4038 126-4075 126-4075 #126-4075 126-4076 126-4076
 #126-4076

T\$TEMP = 000000

#6-507 6-507 #7-522 7-522 7-522 #7-522 7-522 7-522 #7-522 7-522 7-522
 7-522 7-522 #7-522 7-522 7-522 #7-522 7-522 7-522 #7-522 7-522 7-522
 #8-552 8-552 #9-578 9-578 #9-580 9-580 #74-2744 74-2744 #84-2948
 84-2948 #84-2961 84-2961 #88-3073 88-3073 #89-3176 89-3176 #89-3192 89-3192
 #91-3315 91-3315 #93-3346 93-3346 #95-3395 95-3395 #97-3433 97-3433 #99-3478
 99-3478 #102-3545 102-3545 #102-3552 102-3552 #104-3559 104-3559 #104-3579 104-3579
 #106-3589 106-3589 #106-3621 106-3621 #108-3629 108-3629 #108-3653 108-3653 #110-3660
 110-3660 #110-3684 110-3684 #112-3691 112-3691 #112-3715 112-3715 #114-3722 114-3722
 #114-3746 114-3746 #116-3758 116-3758 #116-3868 116-3868 #120-3919 120-3919 #122-3998
 122-3998 #122-3998 122-3998 #122-3998 122-3998 #122-3999 122-3999 #122-3999 122-3999
 #122-3999 122-3999 #122-4000 122-4000 #122-4000 122-4000 #122-4000 122-4000 #122-4001
 122-4001 #122-4001 122-4001 #122-4001 122-4001 #122-4002 122-4002 #122-4002 122-4002
 #122-4002 122-4002 #122-4010 122-4010 #124-4039 124-4039 #124-4039 124-4039 #124-4039
 124-4039 #124-4040 124-4040 #124-4040 124-4040 #124-4040 124-4040 #124-4041 124-4041
 #124-4041 124-4041 #124-4041 124-4041 #124-4042 124-4042 #124-4042 124-4042 #124-4042
 124-4042 #124-4043 124-4043 #124-4043 124-4043 #124-4043 124-4043 #124-4043 124-4043
 #124-4044 124-4044 #124-4044 124-4044 #124-4045 124-4045 #124-4045 124-4045 #124-4045
 124-4045 #124-4052 124-4052 #126-4073 126-4073 126-4073 #104-3557 104-3557 106-3587 #106-3587
 #5-451 102-3543 #102-3543 102-3543 104-3557 #110-3658 110-3658 112-3689 #112-3689
 106-3587 108-3627 #108-3627 108-3627 110-3658 #110-3658 110-3658 116-3751 #116-3751
 112-3689 114-3720 #114-3720 114-3720 116-3751 #116-3751 116-3751 126-4072 126-4072

T\$TEST = 000010

#5-451
 47-1736 49-1823 49-1824 49-1826 52-1947 62-2252 68-2480 68-2485
 68-2488 68-2490 68-2504 68-2511 68-2519 72-2668 72-2682 72-2688 72-2695
 72-2706 72-2710 72-2714 74-2734 74-2737 74-2740 74-2744 82-2879 82-2894
 82-2895 82-2897 82-2935 82-2937 84-2969 84-2970 86-3019 86-3020 88-3054
 88-3059 88-3061 89-3132 89-3134 89-3135 89-3136 89-3137 89-3152 89-3153
 89-3163 89-3176 91-3207 91-3227 91-3228 91-3234 91-3247 91-3248 91-3285
 91-3315 93-3334 93-3345 93-3346 93-3354 93-3357 95-3374 95-3395 97-3413
 97-3433 99-3478 102-3545 102-3552 104-3559 104-3579 106-3589 106-3621 108-3629
 108-3653 110-3660 110-3684 112-3691 112-3715 114-3722 114-3746 116-3758 116-3848
 116-3855 116-3857 116-3859 116-3868

T\$TSTS = 000001

#5-451 #102-3543 #104-3557 #106-3587 #108-3627 #110-3658 #112-3689 #114-3720 #116-3751

T\$SAU = 010013
 T\$SAUT = 010010
 T\$SCLE = 010011
 T\$SDAT = 010030
 T\$SDU = 010012
 T\$SHAR = 010024

#99-3455 99-3478
 #93-3332 93-3346
 #95-3369 95-3395
 #126-4076 126-4076 126-4081
 #97-3405 97-3433
 #122-3995 122-3995 122-4010

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
TSSH	=	010001	#8-539 8-539 8-552
TSSINI	=	010007	#91-3202 91-3315
TSSMSG	=	010003	#74-2727 74-2744
TSSPC	=	000001	#126-4075 126-4082
TSSPRO	=	010000	#6-503
TSSPTA	=	010027	#126-4075 126-4076 #126-4076
TSSRPT	=	010006	#89-3124 89-3176
TSSSOF	=	010025	#124-4038 124-4038 124-4052
TSSSRV	=	010005	#84-2944 84-2948 #84-2952 84-2961
TSSSW	=	010002	#9-561 9-561 9-578
TSSSTES	=	010023	#102-3543 102-3545 102-3552 #104-3557 104-3559 104-3579 #106-3587 106-3589 106-3621 #108-3627 108-3629 108-3653 #110-3658 110-3660 110-3684 #112-3689 112-3691 112-3715 #114-3720 114-3722 114-3746 #116-3751 116-3758 116-3868
T1		017150	G 7-522 #102-3543
T1TRY	=	000146	G #17-945
T2		017352	G 7-522 #104-3557
T3		017624	G 7-522 #106-3587
T4		021230	G 7-522 #108-3627
T4TRY	=	000132	G #17-939
T5		022220	G 7-522 #110-3658
T6		023004	G 7-522 #112-3689
T7		023774	G 7-522 #114-3720
T8		024560	G 7-522 #116-3751
UAM	=	000200	G #10-640
UNIT		013226	G 74-2734 #74-2745
UNITNO		025640	*116-3847 116-3855 116-3857 116-3859 *116-3863 #116-3875
UNREC		012070	68-2519 #68-2544
UNSUC		011466	68-2484 #68-2501
UNXPCT		007624	#58-2106
WAIT		014350	82-2903 82-2911 82-2918 #84-2965 84-2972
WHCHDR		013504	G 50-1906 50-1920 72-2663 #76-2763
WRLOCK	=	000026	G #11-670 70-2619 72-2691
WRTNO	=	000110	G #16-918 *50-1922 89-3153 91-3254 91-3256
WRTN1	=	000112	G #16-919 *50-1925 89-3163
XFNSND		006636	50-1873 #50-1878
XMDB	=	000030	G #16-898 52-1955 82-2888 84-2947 *91-3271
XMSR	=	000026	G #16-897 52-1944 82-2875 82-2881 82-2892 82-2902 82-2910 82-2933 84-2946
XSCNT	=	000036	G *91-3266 #16-901 *102-3547 *104-3565 *106-3596 *106-3596 *106-3596 *106-3596 *108-3639 *108-3639 *108-3639 *112-3701 *112-3701 *112-3701 *116-3776 *116-3806 *116-3833
XSFLG	=	000034	G #16-900 50-1891 56-2001 66-2348 *102-3547 *104-3565 *106-3596 *106-3596 *106-3596 106-3596 106-3597 106-3597 *108-3639 *108-3639 *108-3639 108-3639 110-3670 110-3670 *112-3701 *112-3701 *112-3701 112-3701 114-3732 114-3732 *116-3775 *116-3805 *116-3832
XSPKMM	=	000032	G #16-899 50-1887 *50-1887 56-2009 58-2083 *58-2087 *58-2100 *58-2106 *58-2122 *58-2146 66-2347 *102-3547 *104-3565 *106-3596 *106-3596 *106-3596 *106-3596 *106-3597 *106-3597 *108-3639 *108-3639 *108-3639 *108-3639 *110-3670 *110-3670 *110-3670 *112-3701 *112-3701 *112-3701 *112-3701 *114-3732 *114-3732 *116-3777 *116-3807 *116-3834
XSPTR	=	000106	G #16-917 *50-1893 58-2088 *58-2089 58-2090 *58-2153
X\$ALWA	=	000000	#5-451
X\$FALS	=	000040	#5-451
X\$OFFS	=	000400	#5-451
X\$TRUE	=	000020	#5-451

MACRO NAME	REFERENCES	#122-4001	#122-4002	#124-4040	#124-4041	#124-4042	#124-4043	#124-4045		
GPRML	#122-4000									
HEADER	5-494									
LASTAD	126-4072									
MSBYTE	#5-494	#5-494	#5-494	#5-494						
MSCHEC	#102-3545	102-3545	#104-3559	104-3559	#106-3589	106-3589	#108-3629	108-3629	#110-3660	110-3660
	#112-3691	112-3691	#114-3722	114-3722	#116-3758	116-3758				
MSCNTO	#122-3998	122-3998	#122-3999	122-3999	#122-4000	122-4000	#122-4001	122-4001	#122-4002	122-4002
	#124-4039	124-4039	#124-4040	124-4040	#124-4041	124-4041	#124-4042	124-4042	#124-4043	124-4043
	#124-4044	124-4044	#124-4045	124-4045						
MSCOUN	#68-2480	68-2480	#68-2485	68-2485	#68-2488	68-2488	#68-2490	68-2490	#68-2504	68-2504
	#68-2511	68-2511	#68-2519	68-2519	#74-2734	74-2734	74-2734	74-2734	#74-2737	74-2737
	74-2737	74-2737	74-2737	#74-2740	74-2740	#86-3020	86-3020	#88-3054	88-3054	#88-3059
	88-3059	#88-3061	88-3061	#89-3134	89-3134	#89-3136	89-3136	#89-3152	89-3152	#89-3153
	89-3153	89-3153	89-3153	89-3153	89-3153	89-3153	89-3153	89-3153	#89-3163	89-3163
	89-3163	89-3163	89-3163	89-3163	89-3163	89-3163	89-3163	#93-3354	93-3354	#97-3413
	97-3413	#116-3848	116-3848	#116-3855	116-3855	#116-3857	116-3857	#116-3859	116-3859	
MSDATA	#5-494	5-494	5-494	5-494	5-494	5-494	5-494	5-494	5-494	5-494
	5-494	5-494	5-494	5-494	5-494	5-494	#5-494	5-494	5-494	5-494
	5-494	5-494	5-494	5-494	5-494	5-494	5-494	5-494	5-494	5-494
	5-494	5-494	5-494	5-494	5-494	5-494	5-494	5-494	5-494	5-494
	5-496	#21-1014	21-1014							#5-496
MSDECR	#6-507	6-507	#8-552	8-552	#9-578	9-578	#9-580	9-580	#74-2744	74-2744
	#84-2948	84-2948	#84-2961	84-2961	#88-3073	88-3073	#89-3176	89-3176	#89-3192	89-3192
	#91-3315	91-3315	#93-3346	93-3346	#95-3395	95-3395	#97-3433	97-3433	#99-3478	99-3478
	#102-3552	102-3552	#104-3579	104-3579	#106-3621	106-3621	#108-3653	108-3653	#110-3684	110-3684
	#112-3715	112-3715	#114-3746	114-3746	#116-3868	116-3868	#120-3919	120-3919	#122-4010	122-4010
	#124-4052	124-4052	#126-4073	126-4073	#126-4076	126-4076				
MSDEFA	#122-3998	#122-3998	#122-3999	#122-3999	#122-4000	#122-4000	#122-4001	#122-4001	#122-4002	#122-4002
	#124-4039	#124-4039	#124-4040	#124-4040	#124-4041	#124-4041	#124-4042	#124-4042	#124-4043	#124-4043
	#124-4044	#124-4044	#124-4045	#124-4045						
MSENDE	#8-552	#9-578	#9-580	#74-2744	#84-2948	#84-2961	#88-3073	#89-3176	#89-3192	#91-3315
	#93-3346	#95-3395	#97-3433	#99-3478	#102-3552	#104-3579	#106-3621	#108-3653	#110-3684	#112-3715
	#114-3746	#116-3868	#120-3919	#122-4010	#124-4052	#126-4073				
MSERRI	#49-1823	#49-1823	#72-2668	#72-2668	#72-2682	#72-2682	#72-2688	#72-2688	#72-2695	#72-2695
	#72-2706	#72-2706	#72-2710	#72-2710	#86-3019	#86-3019	#91-3227	#91-3227	#91-3247	#91-3247
MSEXCP	#122-3998	122-3998	122-3998	#122-3999	122-3999	122-3999	#124-4039	124-4039	124-4039	#124-4044
	124-4044	124-4044								
MSEXIT	#102-3545	102-3545	#104-3559	104-3559	#106-3589	106-3589	#108-3629	108-3629	#110-3660	110-3660
	#112-3691	112-3691	#114-3722	114-3722	#116-3758	116-3758				
MSEXSE	#102-3545	#104-3559	#106-3589	#108-3629	#110-3660	#112-3691	#114-3722	#116-3758		
MSEXTJ	#102-3545	#104-3559	#106-3589	#108-3629	#110-3660	#112-3691	#114-3722	#116-3758		
MSGEN	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494	#5-494
	#7-522	#7-522	#8-539	#8-539	#8-539	#8-539	#8-552	#8-552	#9-561	#9-561
	#9-561	#9-561	#9-578	#9-578	#21-1014	#21-1014	#74-2727	#74-2727	#74-2744	#74-2744
	#84-2944	#84-2944	#84-2948	#84-2948	#84-2952	#84-2952	#84-2961	#84-2961	#89-3124	#89-3124

MACRO CROSS REFERENCE
MACRO NAME

REFERENCES

#72-2710	72-2710	#72-2714	#72-2714	72-2714	#74-2734	#74-2734	74-2734	74-2734	#74-2734
74-2734	#74-2734	74-2734	74-2734	#74-2734	74-2734	#74-2734	74-2734	74-2734	#74-2734
74-2734	74-2734	#74-2737	#74-2737	74-2737	74-2737	#74-2737	74-2737	74-2737	#74-2737
74-2737	74-2737	#74-2737	74-2737	#74-2737	74-2737	#74-2737	74-2737	74-2737	#74-2737
74-2737	74-2737	#74-2740	#74-2740	74-2740	#74-2740	74-2740	#74-2740	74-2740	74-2740
#74-2740	74-2740	74-2740	#74-2744	74-2744	#82-2879	82-2879	#82-2894	82-2894	#82-2894
82-2894	#82-2895	#82-2895	82-2895	#82-2895	82-2895	#82-2895	82-2895	#82-2895	82-2895
#82-2895	82-2895	82-2895	#82-2897	#82-2897	82-2897	#82-2897	82-2897	#82-2897	82-2897
#82-2897	82-2897	#82-2897	82-2897	82-2897	#82-2935	82-2935	#82-2935	82-2935	#82-2937
82-2937	#82-2937	82-2937	#84-2948	84-2948	#84-2961	84-2961	#84-2969	84-2969	#84-2970
84-2970	#86-3019	#86-3019	86-3019	#86-3019	86-3019	#86-3019	86-3019	#86-3019	86-3019
#86-3020	#86-3020	86-3020	#86-3020	86-3020	#86-3020	86-3020	86-3020	#86-3020	86-3020
86-3020	#88-3054	#88-3054	88-3054	88-3054	#88-3054	88-3054	#88-3054	88-3054	88-3054
#88-3054	88-3054	88-3054	#88-3059	#88-3059	88-3059	#88-3059	88-3059	88-3059	#88-3059
88-3059	88-3059	#88-3061	#88-3061	88-3061	#88-3061	88-3061	#88-3061	88-3061	88-3061
88-3061	#89-3132	89-3132	#89-3134	#89-3134	89-3134	#89-3134	89-3134	#89-3134	89-3134
89-3134	89-3134	#89-3135	89-3135	#89-3136	#89-3136	89-3136	#89-3136	89-3136	89-3136
#89-3136	89-3136	89-3136	#89-3137	89-3137	#89-3152	#89-3152	89-3152	#89-3152	89-3152
#89-3152	89-3152	89-3152	#89-3152	89-3152	89-3152	#89-3153	#89-3153	89-3153	#89-3153
89-3153	#89-3153	89-3153	#89-3153	89-3153	#89-3153	89-3153	#89-3153	89-3153	#89-3153
#89-3153	89-3153	#89-3153	89-3153	#89-3153	89-3153	#89-3153	89-3153	#89-3153	89-3153
89-3153	89-3153	#89-3163	#89-3163	89-3163	#89-3163	89-3163	#89-3163	89-3163	#89-3163
89-3163	#89-3163	89-3163	89-3163	#89-3163	89-3163	#89-3163	89-3163	#89-3163	89-3163
#89-3163	89-3163	#89-3163	89-3163	89-3163	#89-3163	89-3163	#89-3163	89-3163	89-3163
#91-3207	91-3207	#91-3207	91-3207	#91-3208	91-3208	#91-3227	#91-3227	91-3227	#91-3227
91-3227	#91-3227	91-3227	#91-3227	91-3227	#91-3228	91-3228	#91-3234	91-3234	#91-3234
91-3234	#91-3234	91-3234	#91-3235	91-3235	#91-3247	#91-3247	91-3247	#91-3247	91-3247
#91-3247	91-3247	#91-3247	91-3247	#91-3248	91-3248	#91-3285	91-3285	#91-3285	91-3285
#91-3315	91-3315	#93-3334	#93-3334	93-3334	#93-3334	93-3334	#93-3334	93-3334	#93-3334
93-3334	#93-3334	93-3334	93-3334	#93-3345	93-3345	#93-3345	93-3345	#93-3346	93-3346
#93-3354	#93-3354	93-3354	#93-3354	93-3354	93-3354	#93-3354	93-3354	93-3354	#93-3357
#93-3357	93-3357	#95-3374	95-3374	#95-3395	95-3395	#97-3413	#97-3413	97-3413	#97-3413
97-3413	#97-3413	97-3413	97-3413	#97-3413	97-3413	97-3413	#97-3433	97-3433	#99-3478
99-3478	#102-3545	102-3545	#102-3545	102-3545	#102-3552	102-3552	#104-3559	104-3559	#104-3559
104-3559	#104-3579	104-3579	#106-3589	106-3589	#106-3589	106-3589	#106-3621	106-3621	#108-3629
108-3629	#108-3629	108-3629	#108-3653	108-3653	#110-3660	110-3660	#110-3660	110-3660	#110-3684
110-3684	#112-3691	112-3691	#112-3691	112-3691	#112-3715	112-3715	#114-3722	114-3722	#114-3722
114-3722	#114-3746	114-3746	#116-3758	116-3758	#116-3758	116-3758	#116-3848	#116-3848	116-3848
#116-3848	116-3848	116-3848	#116-3848	116-3848	116-3848	#116-3855	#116-3855	116-3855	#116-3855
116-3855	#116-3855	116-3855	116-3855	#116-3855	116-3855	116-3855	#116-3857	#116-3857	116-3857
#116-3857	116-3857	#116-3857	116-3857	116-3857	#116-3857	116-3857	116-3857	#116-3859	#116-3859
116-3859	#116-3859	116-3859	#116-3859	116-3859	116-3859	#116-3859	116-3859	116-3859	#116-3868
116-3868	#122-3995	122-3995	#122-3998	122-3998	122-3998	122-3998	122-3998	#122-3999	122-3999
122-3999	122-3999	122-3999	#122-4000	122-4000	122-4000	122-4000	#122-4001	122-4001	122-4001
122-4001	#122-4002	122-4002	122-4002	122-4002	#122-4010	122-4010	#124-4038	124-4038	#124-4039
124-4039	124-4039	124-4039	124-4039	124-4039	#124-4040	124-4040	124-4040	124-4040	#124-4041
124-4041	124-4041	124-4041	#124-4042	124-4042	124-4042	124-4042	#124-4043	124-4043	124-4043
124-4043	#124-4044	124-4044	124-4044	124-4044	124-4044	124-4044	#124-4045	124-4045	124-4045
124-4045	#124-4052	124-4052	#126-4072	126-4072	126-4072	126-4072	#126-4076	#126-4076	126-4076
126-4076									

MSGNTA

#8-552	#8-552	#9-578	#9-578	#74-2744	#74-2744	#84-2948	#84-2948	#84-2961	#84-2961
#89-3176	#89-3176	#91-3315	#91-3315	#93-3346	#93-3346	#95-3395	#95-3395	#97-3433	#97-3433

MACRO CROSS REFERENCE
MACRO NAME

REFERENCES

	#99-3478	#99-3478	#102-3552	#102-3552	#104-3579	#104-3579	#106-3621	#106-3621	#108-3653	#108-3653
	#110-3684	#110-3684	#112-3715	#112-3715	#114-3746	#114-3746	#116-3868	#116-3868	#122-4010	#122-4010
	#124-4052	#124-4052	#126-4076	#126-4076	#126-4081	#126-4081				
MSGNTE	#102-3543	#102-3543	#104-3557	#104-3557	#106-3587	#106-3587	#108-3627	#108-3627	#110-3658	#110-3658
	#112-3689	#112-3689	#114-3720	#114-3720	#116-3751	#116-3751				
MSHAPT	#5-494	#5-494								
MSHNAP	#5-494	#5-494								
MSINCR	#5-477	#5-477	#6-503	#6-503	#6-503	#6-503	#8-539	#8-539	#8-539	#8-539
	#9-561	#9-561	#9-561	#9-561	#10-633	#10-633	#47-1736	#49-1823	#49-1824	#49-1826
	#52-1947	#62-2252	#68-2480	#68-2485	#68-2488	#68-2490	#68-2504	#68-2511	#68-2519	#72-2668
	#72-2682	#72-2688	#72-2695	#72-2706	#72-2710	#72-2714	#74-2727	#74-2727	#74-2727	#74-2727
	#74-2734	#74-2737	#74-2740	#74-2744	#82-2879	#82-2894	#82-2895	#82-2897	#82-2935	#82-2937
	#84-2944	#84-2944	#84-2944	#84-2944	#84-2952	#84-2952	#84-2952	#84-2952	#84-2969	#84-2970
	#86-3019	#86-3020	#88-3054	#88-3059	#88-3061	#89-3117	#89-3117	#89-3124	#89-3124	#89-3124
	#89-3124	#89-3132	#89-3134	#89-3135	#89-3136	#89-3137	#89-3152	#89-3153	#89-3163	#89-3176
	#91-3202	#91-3202	#91-3202	#91-3202	#91-3207	#91-3227	#91-3228	#91-3234	#91-3247	#91-3248
	#91-3285	#91-3315	#93-3332	#93-3332	#93-3332	#93-3332	#93-3334	#93-3345	#93-3346	#93-3354
	#93-3357	#95-3369	#95-3369	#95-3369	#95-3369	#95-3374	#95-3395	#97-3405	#97-3405	#97-3405
	#97-3405	#97-3413	#97-3433	#99-3455	#99-3455	#99-3455	#99-3455	#99-3478	#102-3540	#102-3540
	#102-3543	#102-3543	#102-3543	#102-3543	#102-3543	#102-3543	#102-3545	#102-3552	#104-3557	#104-3557
	#104-3557	#104-3557	#104-3557	#104-3557	#104-3559	#104-3579	#106-3587	#106-3587	#106-3587	#106-3587
	#106-3587	#106-3587	#106-3589	#106-3621	#108-3627	#108-3627	#108-3627	#108-3627	#108-3627	#108-3627
	#108-3629	#108-3653	#110-3658	#110-3658	#110-3658	#110-3658	#110-3658	#110-3658	#110-3660	#110-3684
	#112-3689	#112-3689	#112-3689	#112-3689	#112-3689	#112-3689	#112-3691	#112-3715	#114-3720	#114-3720
	#114-3720	#114-3720	#114-3720	#114-3720	#114-3722	#114-3746	#116-3751	#116-3751	#116-3751	#116-3751
	#116-3751	#116-3751	#116-3758	#116-3848	#116-3855	#116-3857	#116-3859	#116-3868	#122-3984	#122-3984
	#122-3995	#122-3995	#122-3995	#122-3995	#124-4038	#124-4038	#124-4038	#124-4038	#126-4075	#126-4075
	#126-4076	#126-4076	#126-4076	#126-4076						
MSLDRO	#72-2714	72-2714	#82-2894	82-2894	#82-2935	82-2935	#82-2937	82-2937	#91-3207	91-3207
	#91-3234	91-3234	#93-3345	93-3345	#93-3357	93-3357				
MSMCHI	#5-451	#5-451								
MSMCLO	#5-451	#5-451								
MSPOP	#6-507	6-507	#8-552	8-552	#9-578	9-578	#9-580	9-580	#74-2744	74-2744
	#84-2948	84-2948	#84-2961	84-2961	#88-3073	88-3073	#89-3176	89-3176	#89-3192	89-3192
	#91-3315	9-3315	#93-3346	93-3346	#95-3395	95-3395	#97-3433	97-3433	#99-3478	99-3478
	#102-3552	102-3552	#104-3579	104-3579	#106-3621	106-3621	#108-3653	108-3653	#110-3684	110-3684
	#112-3715	112-3715	#114-3746	114-3746	#116-3868	116-3868	#120-3919	120-3919	#122-4010	122-4010
	#124-4052	124-4052	#126-4073	126-4073						
MSPRIN	#68-2480	#68-2480	#68-2485	#68-2485	#68-2488	#68-2488	#68-2490	#68-2490	#68-2504	#68-2504
	#68-2511	#68-2511	#68-2519	#68-2519	#74-2734	#74-2734	#74-2737	#74-2737	#74-2740	#74-2740
	#86-3020	#86-3020	#88-3054	#88-3054	#88-3059	#88-3059	#88-3061	#88-3061	#89-3134	#89-3134
	#89-3136	#89-3136	#89-3152	#89-3152	#89-3153	#89-3153	#89-3163	#89-3163	#93-3354	#93-3354
	#97-3413	#97-3413	#116-3848	#116-3848	#116-3855	#116-3855	#116-3857	#116-3857	#116-3859	#116-3859
MSPUSH	#5-477	#5-477	#6-503	#6-503	#8-539	#8-539	#9-561	#9-561	#10-633	#10-633
	#74-2727	#74-2727	#84-2944	#84-2944	#84-2952	#84-2952	#89-3117	#89-3117	#89-3124	#89-3124
	#91-3202	#91-3202	#93-3332	#93-3332	#95-3369	#95-3369	#97-3405	#97-3405	#99-3455	#99-3455
	#102-3540	#102-3540	#102-3543	#102-3543	#104-3557	#104-3557	#106-3587	#106-3587	#108-3627	#108-3627
	#110-3658	#110-3658	#112-3689	#112-3689	#114-3720	#114-3720	#116-3751	#116-3751	#122-3984	#122-3984
	#122-3995	#122-3995	#124-4038	#124-4038						
MSPUT	#68-2480	#68-2480	#68-2480	#68-2480	#68-2485	#68-2485	#68-2485	#68-2485	#68-2488	#68-2488
	#68-2488	#68-2490	#68-2490	#68-2490	#68-2504	#68-2504	#68-2504	#68-2511	#68-2511	#68-2511
	#68-2511	#68-2519	#68-2519	#68-2519	#74-2734	#74-2734	#74-2734	#74-2734	#74-2734	#74-2734

MACRO NAME	REFERENCES									
MSPUT1	#74-2737	#74-2737	#74-2737	#74-2737	#74-2737	#74-2737	#74-2737	#74-2740	#74-2740	#74-2740
	#74-2740	#82-2895	#82-2895	#82-2895	#82-2895	#82-2895	#82-2895	#82-2897	#82-2897	#82-2897
	#82-2897	#86-3020	#86-3020	#86-3020	#86-3020	#86-3020	#88-3054	#88-3054	#88-3054	#88-3054
	#88-3059	#88-3059	#88-3061	#88-3061	#88-3061	#89-3134	#89-3134	#89-3134	#89-3136	#89-3136
	#89-3136	#89-3152	#89-3152	#89-3152	#89-3152	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153
	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	#89-3163	#89-3163	#89-3163	#89-3163
	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#93-3334	#93-3334	#93-3334
	#93-3334	#93-3334	#93-3354	#93-3354	#93-3354	#97-3413	#97-3413	#97-3413	#97-3413	#116-3848
	#116-3848	#116-3848	#116-3855	#116-3855	#116-3855	#116-3855	#116-3857	#116-3857	#116-3857	#116-3857
	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859
	#68-2480	#68-2480	#68-2480	#68-2480	#68-2480	#68-2480	#68-2485	#68-2485	#68-2485	#68-2485
	#68-2485	#68-2485	#68-2488	#68-2488	#68-2488	#68-2488	#68-2490	#68-2490	#68-2490	#68-2490
	#68-2504	#68-2504	#68-2504	#68-2504	#68-2511	#68-2511	#68-2511	#68-2511	#68-2511	#68-2511
	#68-2519	#68-2519	#68-2519	#68-2519	#74-2734	#74-2734	#74-2734	#74-2734	#74-2734	#74-2734
	#74-2734	#74-2734	#74-2734	#74-2734	#74-2737	#74-2737	#74-2737	#74-2737	#74-2737	#74-2737
	#74-2737	#74-2737	#74-2737	#74-2737	#74-2737	#74-2737	#74-2740	#74-2740	#74-2740	#74-2740
	#74-2740	#74-2740	#82-2895	#82-2895	#82-2895	#82-2895	#82-2895	#82-2895	#82-2895	#82-2895
	#82-2897	#82-2897	#82-2897	#82-2897	#82-2897	#82-2897	#82-2897	#82-2897	#82-2897	#82-2897
	#86-3020	#86-3020	#86-3020	#86-3020	#88-3054	#88-3054	#88-3054	#88-3054	#88-3054	#88-3054
	#88-3059	#88-3059	#88-3059	#88-3059	#88-3061	#88-3061	#88-3061	#88-3061	#89-3134	#89-3134
	#89-3134	#89-3134	#89-3136	#89-3136	#89-3136	#89-3136	#89-3152	#89-3152	#89-3152	#89-3152
#89-3152	#89-3152	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	
#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	#89-3153	
#89-3153	#89-3153	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	
#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	#89-3163	
#89-3163	#89-3163	#93-3334	#93-3334	#93-3334	#93-3334	#93-3334	#93-3334	#93-3334	#93-3334	
#93-3354	#93-3354	#93-3354	#93-3354	#97-3413	#97-3413	#97-3413	#97-3413	#97-3413	#97-3413	
#116-3848	#116-3848	#116-3848	#116-3848	#116-3855	#116-3855	#116-3855	#116-3855	#116-3855	#116-3855	
#116-3857	#116-3857	#116-3857	#116-3857	#116-3857	#116-3857	#116-3859	#116-3859	#116-3859	#116-3859	
#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	#116-3859	
MSRADI	#122-3998	#122-3998	#122-3999	#122-3999	#122-4000	#122-4000	#122-4001	#122-4001	#122-4002	#122-4002
	#124-4039	#124-4039	#124-4040	#124-4040	#124-4041	#124-4041	#124-4042	#124-4042	#124-4043	#124-4043
	#124-4044	#124-4044	#124-4045	#124-4045						
MSRNRO	#91-3234	#91-3234	#91-3285	#91-3285						
	#5-477	#5-477	#6-503	#6-503	#8-539	#8-539	#9-561	#9-561	#10-633	#10-633
MSSETS	#74-2727	#74-2727	#84-2944	#84-2944	#84-2952	#84-2952	#89-3117	#89-3117	#89-3124	#89-3124
	#91-3202	#91-3202	#93-3332	#93-3332	#95-3369	#95-3369	#97-3405	#97-3405	#99-3455	#99-3455
	#102-3540	#102-3540	#102-3543	#102-3543	#104-3557	#104-3557	#106-3587	#106-3587	#108-3627	#108-3627
	#110-3658	#110-3658	#112-3689	#112-3689	#114-3720	#114-3720	#116-3751	#116-3751	#122-3984	#122-3984
	#122-3995	#122-3995	#124-4038	#124-4038						
	#47-1736	47-1736	49-1823	#49-1824	49-1824	#49-1826	49-1826	#52-1947	52-1947	#62-2252
	62-2252	#68-2480	68-2480	#68-2485	68-2485	#68-2488	68-2488	#68-2490	68-2490	#68-2504
68-2504	#68-2511	68-2511	#68-2519	68-2519	72-2668	72-2682	72-2688	72-2695	72-2706	
72-2710	#72-2714	72-2714	#74-2734	74-2734	#74-2737	74-2737	#74-2740	74-2740	#74-2744	
74-2744	#82-2879	82-2879	#82-2894	82-2894	#82-2895	82-2895	#82-2897	82-2897	#82-2935	
82-2935	#82-2937	82-2937	#84-2969	84-2969	#84-2970	84-2970	86-3019	#86-3020	86-3020	
#88-3054	88-3054	#88-3059	88-3059	#88-3061	88-3061	#89-3132	89-3132	#89-3134	89-3134	
#89-3135	89-3135	#89-3136	89-3136	#89-3137	89-3137	#89-3152	89-3152	#89-3153	89-3153	
#89-3163	89-3163	#89-3176	89-3176	#91-3207	91-3207	91-3227	#91-3228	91-3228	#91-3234	
91-3234	91-3247	#91-3248	91-3248	#91-3285	91-3285	#91-3315	91-3315	#93-3334	93-3334	
#93-3345	93-3345	#93-3346	93-3346	#93-3354	93-3354	#93-3357	93-3357	#95-3374	95-3374	
#95-3395	95-3395	#97-3413	97-3413	#97-3433	97-3433	#99-3478	99-3478	#102-3545	102-3545	

MACRO CROSS REFERENCE

MACRO NAME

REFERENCES

MSTLAB

MSTSTL

MSWORD

POINTE POP

PRINTB
PRINTF
PRINTS
PRINTX
PUSH

#102-3552	102-3552	#104-3559	104-3559	#104-3579	104-3579	#106-3589	106-3589	#106-3621	106-3621
#108-3629	108-3629	#108-3653	108-3653	#110-3660	110-3660	#110-3684	110-3684	#112-3691	112-3691
#112-3715	112-3715	#114-3722	114-3722	#114-3746	114-3746	#116-3758	116-3758	#116-3848	116-3848
#116-3855	116-3855	#116-3857	116-3857	#116-3859	116-3859	#116-3868	116-3868		
#47-1736	#49-1823	#49-1824	#49-1826	#52-1947	#62-2252	#68-2480	#68-2485	#68-2488	#68-2490
#68-2504	#68-2511	#68-2519	#72-2668	#72-2682	#72-2688	#72-2695	#72-2706	#72-2710	#72-2714
#74-2734	#74-2737	#74-2740	#74-2744	#82-2879	#82-2894	#82-2895	#82-2897	#82-2935	#82-2937
#84-2969	#84-2970	#86-3019	#86-3020	#88-3054	#88-3059	#88-3061	#89-3132	#89-3134	#89-3135
#89-3136	#89-3137	#89-3152	#89-3153	#89-3163	#89-3176	#91-3207	#91-3227	#91-3228	#91-3234
#91-3247	#91-3248	#91-3285	#91-3315	#93-3334	#93-3345	#93-3346	#93-3354	#93-3357	#95-3374
#95-3395	#97-3413	#97-3433	#99-3478	#102-3545	#102-3552	#104-3559	#104-3579	#106-3589	#106-3621
#108-3629	#108-3653	#110-3660	#110-3684	#112-3691	#112-3715	#114-3722	#114-3746	#116-3758	#116-3848
#116-3855	#116-3857	#116-3859	#116-3868						
#47-1736	47-1736	#49-1823	#49-1823	49-1823	#49-1824	49-1824	#49-1826	49-1826	#52-1947
52-1947	#62-2252	62-2252	#68-2480	68-2480	#68-2485	68-2485	#68-2488	68-2488	#68-2490
68-2490	#68-2504	68-2504	#68-2511	68-2511	#68-2519	68-2519	#72-2668	#72-2668	72-2668
#72-2682	#72-2682	72-2682	#72-2688	#72-2688	72-2688	#72-2695	#72-2695	72-2695	#72-2706
#72-2706	72-2706	#72-2710	#72-2710	72-2710	#72-2714	72-2714	#74-2734	74-2734	#74-2737
74-2737	#74-2740	74-2740	#74-2744	74-2744	#82-2879	82-2879	#82-2894	82-2894	#82-2895
82-2895	#82-2897	82-2897	#82-2935	82-2935	#82-2937	82-2937	#84-2969	84-2969	#84-2970
84-2970	#86-3019	#86-3019	86-3019	#86-3020	86-3020	#88-3054	88-3054	#88-3059	88-3059
#88-3061	88-3061	#89-3132	89-3132	#89-3134	89-3134	#89-3135	89-3135	#89-3136	89-3136
#89-3137	89-3137	#89-3152	89-3152	#89-3153	89-3153	#89-3163	89-3163	#89-3176	89-3176
#91-3207	91-3207	#91-3227	#91-3227	91-3227	#91-3228	91-3228	#91-3234	91-3234	#91-3247
#91-3247	91-3247	#91-3248	91-3248	#91-3285	91-3285	#91-3315	91-3315	#93-3334	93-3334
#93-3345	93-3345	#93-3346	93-3346	#93-3354	93-3354	#93-3357	93-3357	#95-3374	95-3374
#95-3395	95-3395	#97-3413	97-3413	#97-3433	97-3433	#99-3478	99-3478	#102-3545	102-3545
#102-3552	102-3552	#104-3559	104-3559	#104-3579	104-3579	#106-3589	106-3589	#106-3621	106-3621
#108-3629	108-3629	#108-3653	108-3653	#110-3660	110-3660	#110-3684	110-3684	#112-3691	112-3691
#112-3715	112-3715	#114-3722	114-3722	#114-3746	114-3746	#116-3758	116-3758	#116-3848	116-3848
#116-3855	116-3855	#116-3857	116-3857	#116-3859	116-3859	#116-3868	116-3868		
#5-494	#5-494	#7-522	#7-522	#7-522	#7-522	#7-522	#7-522	#7-522	#7-522
#7-522	#7-522	#49-1823	#49-1823	#49-1823	#49-1823	#72-2668	#72-2668	#72-2668	#72-2668
#72-2682	#72-2682	#72-2682	#72-2682	#72-2688	#72-2688	#72-2688	#72-2688	#72-2695	#72-2695
#72-2695	#72-2695	#72-2706	#72-2706	#72-2706	#72-2706	#72-2710	#72-2710	#72-2710	#72-2710
#86-3019	#86-3019	#86-3019	#86-3019	#91-3227	#91-3227	#91-3227	#91-3227	#91-3247	#91-3247
#91-3247	#91-3247	#102-3545	#104-3559	#106-3589	#108-3629	#110-3660	#112-3691	#114-3722	#116-3758
#122-3998	#122-3998	#122-3999	#122-3999	#122-4000	#122-4000	#122-4001	#122-4001	#122-4002	#122-4002
#124-4039	#124-4039	#124-4040	#124-4040	#124-4041	#124-4041	#124-4042	#124-4042	#124-4043	#124-4043
#124-4044	#124-4044	#124-4045	#124-4045	#126-4076	#126-4076				
5-484									
#23-1070	43-1695	43-1696	52-1948	52-1956	58-2138	60-2189	60-2190	68-2531	68-2532
72-2715	72-2716	72-2717	72-2718	74-2742	74-2743	78-2808	78-2809	80-2846	80-2847
86-3024	86-3025	86-3026	88-3062	88-3063	89-3170	89-3171	89-3172	89-3173	89-3174
89-3175	97-3411	97-3412							
#68-2504	#74-2734	#74-2737	#74-2740	#86-3020		116-3848	116-3855	116-3857	116-3859
88-3054	88-3059	88-3061	93-3354	97-3413					
#89-3134	#89-3136	#89-3152	#89-3153	#89-3163					
68-2480	68-2485	68-2488	68-2490	68-2511	68-2519				
#23-1057	43-1688	43-1689	52-1942	52-1946	58-2128	60-2178	60-2179	68-2456	68-2457
72-2652	72-2653	72-2654	72-2655	74-2728	74-2729	78-2784	78-2785	80-2824	80-2825
86-2998	86-2999	86-3000	88-3047	88-3048	89-3125	89-3126	89-3127	89-3128	89-3129

MACRO NAME	REFERENCES							
	89-3130	97-3407	97-3408					
REDEF	#91-3207							
RFLAGS	#91-3285							
SETPRI	82-2894							
SETVEC	82-2895	82-2897	93-3334					
SVC	#5-450	5-451						
SWAPIN	#23-1087	#49-1773	#49-1784	#49-1791	#49-1812			
SWAPOW	#23-1108	50-1869						
TSTID	#35-1512	102-3544	104-3558	106-3588	108-3628	110-3659	112-3690	114-3721
TUREAD	#31-1394	#106-3597	#110-3670	#114-3732				
TURTRY	#29-1319	#106-3596	#106-3597	#108-3639	#110-3670	#112-3701	#114-3732	
TUSEEK	#27-1257	104-3565						
TUSELF	#33-1471	#102-3547						
TUWRIT	#25-1149	106-3596	108-3639	112-3701				