

TU58

PERF EXERCISER
CZTUUDO

AH-E649D-MC
FICHE 1 OF 1

JUL 1982
COPYRIGHT © 79-82
MADE IN USA



Table with multiple columns and rows of data, likely a performance or exercise log. The content is too faint to transcribe accurately.



.REM 8

IDENTIFICATION

PRODUCT CODE: AC-E648D-MC
PRODUCT NAME: CZTUUDO TU58 PERF EXER
PRODUCT DATE: FEBRUARY 1982
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) 1979,1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

.SBTTL REVISION HISTORY

HISTORY

.REM 8

JUNE 18, 1979	INITIAL RELEASE	CZTU 40
JULY 1, 1979	SECOND RELEASE	CZTUUB0
JUNE 1, 1980	THIRD RELEASE	CZTUUB1
OCTOBER 1, 1981	FOURTH RELEASE	CZTUUC0
MARCH 1, 1982	FIFTH RELEASE	CZTUUDO

CZTUUA0

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

CHANGES TO CZTUUA0

1. THE PROGRAM WAS MODIFIED TO RUN UNDER THE NEW DIAGNOSTIC SUPERVISOR CHSAA0. AS A RESULT OF THIS CONVERSION, THIS PROGRAM NOW OPERATES IN 8K AND PAPERTAPE DISTRIBUTION REQUIRES ONLY ONE PART AK-E650B-MC.

CHANGES TO CZTUUB0

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

CHANGES TO CZTUUB1

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

CHANGES TO CZTUUC0

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

TABLE OF CONTENTS

- 1.0 GENERAL INFORMATION
- 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 TUSB 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 PRECEDING) THE UNITS ARE POLLED, AND THEIR ENTIRE RESPONSES EVALUATED ROUND ROBIN. IF ANY ERROR INITIATES A RETRY, THE SCHED-

ULING PROCESS IS MODIFIED TO COMMUNICATE WITH ONLY 1 UNIT UNTIL COMPLETION OF THE RETRY PROCEDURE. THEN, A RETRY BY ANOTHER UNIT MAY PROCEED, OR THE SYSTEM CONTINUES NORMALLY.

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

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

ERROR DESCRIPTIONS:

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

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

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

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE

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

TU58 CONTROLLER AND DRIVE(S). DL, DLV, OR PDT COMPATIBLE 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 5-8 -- ONE MAY SELECT A MINIMUM OF 8, TO
A MAXIMUM OF 512 BLOCKS TO WRITE,
READ; WRITE VERIFY; AND READ REDUCED,
AS EXPLAINED IN SECTION 6.0.

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

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

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

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

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

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

3.0 ERROR INFORMATION -----

ERROR INFORMATION IS PROVIDED ON 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. THIS TEST IS IN
512 BYTE/BLOCK MODE.

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

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

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

TEST 5: WRITE TAPE

TEST 6: READ TAPE

TEST 7: 'WRITE VERIFY' TAPE

TEST 8: READ MODIFIED THRESHOLD TAPE

TEST 9: THE FIRST PART OF TEST 9 DETERMINES IF A UNIT IS
CAPABLE OF MODIFIED RADIAL SERIAL PROTOCOL. THIS
PART OF THE TEST IS WRITTEN USING RADIAL SERIAL
PROTOCOL, AND DETERMINES THE PROTOCOL OF A UNIT BY

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

PROGRAM HEADER AND TABLES
REVISION HISTORY

MACRO M1113 25-FEB-82 07:06 PAGE 5 1

SEQ 0009

457
458
484
486
487
489
490
491
492
493
494
495
496
497
498
506
507

002000

002000

002000

002000

002000

002000

002001

002002

002003

002004

002005

002006

002007

002010

002010

002011

002011

002012

002012

002014

002014

002016

002016

002020

002020

002022

002022

002024

002024

002026

002026

002030

002030

002032

002032

002034

002034

002036

002036

002040

002040

002042

002042

002044

.TITLE PROGRAM HEADER AND TABLES
.SBTTL PROGRAM HEADER

.ENABL ABS,AMA
= 2000
.NLIST BEX
BGNMOD

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

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

HEADER CZTUU,D,0,3600.,1,PRI07

LSNAME::

.ASCII /C/
.ASCII /Z/
.ASCII /T/
.ASCII /U/
.ASCII /U/
.BYTE 0
.BYTE 0
.BYTE 0

LSREV::

.ASCII /D/

LSDEPO::

.ASCII /O/

LSUNIT::

.WORD TSPTHV

LSIML::

.WORD 3600.

LSHPCP::

.WORD LSHARD

LSSPCP::

.WORD LSSOFT

LSHPTP::

.WORD LSHW

LSSTP::

.WORD LSSW

LSLADP::

.WORD LSLAST

LSSTA::

.WORD 0

LSCO::

.WORD 0

LSDTYP::

.WORD 1

LSAPT::

.WORD 0

LSDTP::

.WORD LSDISPATCH

LSPRIO::

.WORD PRI07

LSENVI::

PROGRAM HEADER AND TABLES
PROGRAM HEADER

MACRO M1113 25-FEB-82 07:06 PAGE 5-1^{K 1}

SEQ 0010

002044 000000
002046
002046 000000
002050
002050 003
002051 003
00205
002052 000000
002054 000000
002056
002056 000000
002060
002060 005510
002062
002062 015150
002064
002064 000000
002066
002066 000000
002070
002070 017302
002072
002072 017156
002074
002074 000000
002076
002076 002122
002100
002100 104035
002102
002102 000000
002104
002104 016164
002106
002106 017136
002110
002110 016754
002112
002112 002142
002114
002114 000000
002116
002116 000000
002120
002120 000000

508
509

DESCRIP <TU58 PERF EXER>

002122
002122
002122 124 125 065

L\$EXP1:: .WORD 0
L\$MREV:: .WORD 0
L\$EF:: .BYTE C\$REVISION
 .BYTE C\$EDIT
L\$SPC:: .WORD 0
 .WORD 0
L\$DEVP:: .WORD 0
L\$REPP:: .WORD L\$DVTYP
L\$EXP4:: .WORD L\$RPT
L\$EXP5:: .WORD 0
L\$AUT:: .WORD 0
L\$DUT:: .WORD L\$AU
L\$LUN:: .WORD L\$DU
L\$DESP:: .WORD 0
L\$LOAD:: .WORD L\$DESC
 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\$DLY:: .WORD 0
L\$HIME:: .WORD 0
L\$DESC:: .ASCIZ /TU58 PERF EXER/
 .EVEN

```
511  
512  
513  
514  
515  
516 002142  
517 002142 000000  
518 002144 177777  
519 002146 177777  
520 002150  
      :++  
      :THE PROTECT TABLE IS USED BY THE MONITOR TO WARN THE OPERATOR WHEN HE  
      :TRIES TO TEST THE LOAD DEVICE.  
      :--  
      BGNPROT  
      :DEVICE CSR  
      :NO MASS BUS  
      :NO DRIVE  
      L$PROT::  
      .WORD 0  
      .WORD -1  
      .WORD -1  
      ENDPROT
```

527
528
529
530
531
532
533
534
535

.SBTTL DISPATCH TABLE

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

DISPATCH 9

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

002150
002150 000011
002152
002152 017304
002154 017506
002156 017760
002160 021352
002162 022756
002164 023746
002166 024532
002170 025522
002172 026306

536

```

:++
: THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: THE TEST-DE 'ICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES.
:--

```

```

        .WORD    L10001-LSHW/2
LSHW::
DFPTBL::

```

```

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

```

L10001:

PROGRAM HEADER AND TABLES
SOFTWARE P-TABLE

MACRO M1113 25-FEB-82 07:06 PAGE 9 2

SEN 0014

```

567      .SBTTL  SOFTWARE P-TABLE
568
569      :++
570      : THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
571      : PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
572      :--
573
574      BGNSW  SFPTBL                                .WORD  L10002-L$SW/2
              002206 000007
              002206
              002210
              002210
575
576      002210 000010      LENGTH: .WORD 8.          :TAPE LENGTH
577      002212 000001      STAEOP: .WORD 1          :PRINT STATISTICS AT EOP
578      002214 000001      PRBUF:  .WORD 1          :PRINT DATA BUF ON COMP. ERROR
579      002216 000001      CMPDAT: .WORD 1          :COMPARE DATA
580      002220 000001      DRVCHK: .WORD 1          :ADD DR # TO DATA
581      002222 000001      EVLTHR: .WORD 1          :THRESHOLD FOR EVL TEST
582      002224 000000      PPSOT9: .WORD 0          :PRINT UNIT PROTOCOL SUMMARY (TST9)
583
584
585
586
587
588
589
590
591      002226      ENDSW                                L10002:
              002226
592
593      002226      ENDMOD

```

```

606          .TITLE GLOBAL AREAS
607          .SBTTL  GLOBAL EQUATES SECTION
635
645          BGNMOD
646 002226
647
648          :++
649          : THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
650          : ARE USED IN MORE THAN ONE TEST.
651          :--
652
653          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.CONTINUC== 30. ; CONTINUE COMMAND WAS ISSUED
          000035 EF.NEW== 29. ; A NEW PASS HAS BEEN STARTED
          000034 EF.PWR== 28. ; A POWER-FAIL/POWER-UP OCCUPRED
          :
          : 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

```

667      .SBTTL  ERROR CODE EQUATES
668
669      ;THE ERROR CODE OFFSET VALUES :
670      ;USED BY ROUTINE 'LOG' TO INDEX (BY R5) INTO DEVICE'S DATA BLOCK AND
671      ;INCREMENT STATISTICS.
672
673      000002      SFTRD   ==      2
674      000004      SFTWR   ==      4
675      000006      RCINIT  ==      6
676      000010      OTL     ==      8.
677      000012      OVRN    ==     10.
678      000014      BDCOM   ==     12.
679      000016      HRDRD   ==     14.
680      000020      HRDWR   ==     16.
681      000022      BDCHK   ==     18.
682      000024      SKERR   ==     20.
683      000026      WRLOCK  ==     22.
684      000030      NOMOT   ==     24.
685      000032      CNINIT  ==     26.
686      000034      PARTL   ==     28.
687      000036      NOUNIT  ==     30.
688      000040      CMNDER  ==     32.
689      000042      RECERR  ==     34.
690      000044      SLFER   ==     36.
691      000046      SUCOTL  ==     38.
692      000050      TORCVB  ==     40.
693      000054      NCART   ==     44.
694      000056      TOSNDB  ==     46.
695
696      ;          IN ADDITION, SYSTEM SETUP OR RUNTIME ERRORS ARE:
697
698      ;      100.  -      ALL UNITS ABORTED
699
700      ;      101.  -      MORE THAN 8. UNITS (16 DRIVES) REQUESTED
701
702      ;      102.  -      NEITHER DRIVE SELECTED FOR THIS CONTROLLER
703
704      ;          ALL THE ABOVE ARE CLASSIFIED AS SYSTEM FATAL
705
  
```

```
707 .SBTTL GENERAL EQUATES
708 ;RADIAL SERIAL CODES:
709 -----
710 ;THE FLAG BYTE CODES ARE:
711 RSCMND == 2 ;'COMMAND' PACKET
712 RSCONT == 20 ;'CONTINUE' SINGLE BYTE
713 RSXON == 20 ;'XON' SINGLE BYTE
714 RSXOFF == 23 ;'XOFF' SINGLE BYTE
715 RSINIT == 4 ;'INIT' SINGLE BYTE
716 RSDATA == 1 ;'DATA' PACKET
717 RSEND == RSCMND ;'END' PACKET FLAG IS 'COMMAND'
718 -----
719 ;END PACK SIZE:
720 RSND SZ == 14. ;TOTAL BYTES IN COMMAND PACKET
721 ;MESSAGE PACK SIZE:
722 RSMSIZ == 12 ;10. BYTES FOR BYTE COUNT INSIDE CMND PACK
723 ;DATA PACK SIZE:
724 RSDASZ == 132. ;TOTAL BYTES IN DATA PACKET
725 ;DATA + END PACK SIZE:
726 RSDNSZ == RSDASZ+RSND SZ
727 ;GET CHARACTERISTICS DATA PACKET SIZE
728 RSGCDP == 28. ;TOTAL BYTES FOR GET CHAR DATA PACKET
729 ;MINUS THE END PACKET
730 RSSNSZ == RSMSIZ + 4 ;SIZE FOR SENDING COMMAND PACK
731 RCBFSZ == 4*RSDASZ+RSND SZ ;4 DATA PAKS AND END PACK
732 ;IS SIZE OF RCV BUFFERS
733 -----
734 ; THE OP CODES ARE:
735
736 RSSEND == 100 ;END PACK DESCRIPTOR
737 RSSWR == 3 ;WRITE
738 RSSRD == 2 ;READ
739 RSSSEK == 5 ;SEEK
740 RSSGET == 12 ;GET CHARACTERISTICS
741 RSSNOP == 0 ;NO-OPERATION
742 RSSNIT == 1 ;INITIALIZE
743 RSSSLF == 7 ;SELF TEST
744 -----
745 ;THE SUCCESS CODES ARE:
746
747 ESABO == -48. ;BAD COMMAND FROM HOST
748 ESNCRD == -9. ;NO CARTRIDGE
749 ESNONX == -8. ;NO DRIVE
750 ESOK == 0 ;OP COMPLETE SUCCESS
751 ESPART == -2 ;PARTIAL OP
752 ESSK == -32. ;SEEK ERROR
753 ESTRY == 1 ;RETRY OCCURRED
754 ESWLOC == -11. ;WRITE PROTECTED
755 ESNOMO == -33. ;MOTOR STOPPED
756 ESCMD == -48. ;COMMAND ERROR
757 ESREC == -55. ;BAD RECORD NUMBER.
758 ESCKS == -17. ;TU CHKSUM ERROR
759 ESSLF == -1. ;SELF TEST ERROR
760 ESCKSM=ESCKS
761 ESWR=ESCKS
762 ESRD=ESCKS
763 -----
```

765
766
767
768
769
770 002226 002322
771 002230 003054
772 002232 003114
773 002234 002536
774 002236 003000
775 002240 003260
776 002242 002404
777 002244 003154
778 002246 003216
779 002250 002556
780 002252 002306
781 002254 002514
782 002256 002446
783 002260 002620
784 002262 002634
785 002264 002656
786 002266 002704
787 002270 002720
788 002272 002364
789 002274 002740
790 002276 002764
791 002300 002322
792 002302 002464
793 002304 003032

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

;HERE ARE THE MESSAGES PROPER:

795						
796						
797	002306	123	105	105	MSSKER:: .ASCIZ /SEEK ERROR/	;DEVICE COULD NOT READ HEADER
798					.EVEN	
799	002322	123	131	123	MSNLOG:: .ASCIZ /SYSTEM ERROR/	;DIAGNOSTIC HUNG. BETTER RE-BOOT
800					.EVEN	
801	002340	102	101	104	MSBDA:: .ASCIZ /BAD DATA IN PACKET/	;HOST DATA CHECK FOUND ERROR, DEVICE MAY
802					.EVEN	;HAVE READ CORRECTLY.
803	002364	123	105	114	MSSELF:: .ASCIZ /SELF TEST ERROR/	;MICRO DIAGNOSTIC FAILED, BUT DEVICE COULD STILL
804					.EVEN	;SEND AN END PACKET.
805	002404	102	101	104	MSCOM:: .ASCIZ /BAD DATA W-O DATA CHECK	ERR AT TU/ ;PREVIOUS DATA CHECK
806					.EVEN	;ERROR NOT DUE TO DEVICE READ OPERATION
807	002446	115	117	124	MSNOMO:: .ASCIZ /MOTOR STOPPED/	;DEVICE COULD NOT GET ANY MEANINGFUL SIGNAL
808					.EVEN	;FROM TAPE.
809	002464	103	101	122	MSNCTP:: .ASCIZ /CARTRIDGE NOT IN PLACE/	;NO MEDIA OR BAD SWITCH
810					.EVEN	
811	002514	127	122	111	MSWPRO:: .ASCIZ /WRITE PROTECTION/	;CARTRIDGE WRITE PROTECT TAB MISSING OR
812					.EVEN	;SWITCH BAD
813	002536	122	105	103	MSRNIT:: .ASCIZ /RECIEVING INIT/	;DEVICE SENT INIT REQUEST
814					.EVEN	
815	002556	110	117	123	MSHCHK:: .ASCIZ /HOST FOUND PACKET CHECKSUM ERROR/	;DEVICE SENT PACK WITH
816					.EVEN	;BAD CHECKSUM
817	002620	103	101	116	MSNIT:: .ASCIZ /CAN'T INIT/	;DEVICE SENT BYTE OTHER THAN "CONTINUE"
818					.EVEN	;DURING INITIALIZATION
819	002634	120	101	122	MSPART:: .ASCIZ /PARTIAL OPERATION/	;END OF MEDIUM ENCOUNTERED
820					.EVEN	
821	002656	042	116	117	MSUNIT:: .ASCIZ /'NON-EXISTENT' DRIVE/	;DEVICE RECV'D TOO LARGE DRIVE NUMBER
822					.EVEN	
823	002704	102	101	104	MSCMD:: .ASCIZ /BAD COMMAND/	;DEVICE COULD NOT UNDERSTAND HOST
824					.EVEN	
825	002720	102	101	104	MSREC:: .ASCIZ /BAD RECORD NO./	;DEVICE RECV'D TOO LARGE A RECORD NUMBER
826					.EVEN	
827	002740	127	122	117	MSWRSP:: .ASCIZ /WRONG SUCCESS CODE/	;HOST COULD NOT DECIPHER CODE IN END PACK
828					.EVEN	
829	002764	116	117	040	MSNRSP:: .ASCIZ /NO RESPONSE/	;TIME OUT WAITING FOR BYTE IN RCV BUF ON INTERFACE.
830					.EVEN	
831	003000	111	116	104	MSQRSP:: .ASCIZ \INDECIPHERABLE FLAG BYTE\	;HOST COULD NOT UNDERSTAND 1ST BYTE OF
832					.EVEN	;RESPONSE FROM TU AS PROPER PROTOCOL
833	003032	124	111	115	MSTOSN:: .ASCIZ /TIME OUT ON SEND/	;DLV 'READY' NEVER WENT HIGH
834					.EVEN	
835	003054	122	105	103	MSSFRD:: .ASCIZ /RECOV. DATA CHECK ERR ON RD OP/	;TU58 RESPONDED WITH 'DATA-CHECK'
836					.EVEN	;ERROR ON READ OP. ;HOST RETRY(S) SUCCESSFUL
837	003114	122	105	103	MSSFWR:: .ASCIZ /RECOV. DATA CHECK ERR ON WR OP/	;SAME BUT WR OR WR VERIFY OPERATION
838					.EVEN	
839	003154	125	116	122	MSHDRD:: .ASCIZ /UNRECOV. DATA CHECK ERR ON RD OP/	;TU58 RESPONDED WITH 'DATA-CHECK'
840					.EVEN	;ERROR ON READ OP. ;RETRIES UNSUCCESSFUL
841	003216	125	116	122	MSHDWR:: .ASCIZ /UNRECOV. DATA CHECK ERR ON WR OP/	;SAME BUT WR OPERATION
842					.EVEN	
843	003260	104	114	126	MSOVRN:: .ASCIZ /DLV ERROR IN RECEIVE/	;DLV ERROR (THE CONTENTS PRINTED OUT)
844					.EVEN	

.SBTTL MISC STORAGE AND EQUATES

846				
847				
848				
849	003306	000000	SYSTAT:: .WORD	:SYSTEM STATUS WORD
850				:BIT7-BIT15 = 1ST BYTE OF PACK RCVD
851				:BIT05 = NOT USED
852				:BIT04 = NOT USED
853				:BIT03 = NOT USED
854				:BIT02 =RETRY BAD FLAG BYTE
855				:BIT01 = RETRY OCCURRING
856				:BIT00 = CHKSUM IS ODD (TEMP STORAGE)
857				
858	003310	000000	TAPLEN:: .WORD	:# RECORDS
859	003312	000000	DEVPTR:: .WORD	:>NEXT UNIT ADDRESS
860	003314	000000	RCFLG:: .WORD	:TEMP STORE FOR GTBYTE
861	003316	000000	RCBLNT:: .WORD	:TEMP STORAGE FOR GTBYTE
862	003320	000004	FTLNM:: .WORD 4	:NUMBER OF TRIES BEFORE ABORT
863	003322	000000	DONE:: .WORD	:1 IF TEST ALGORITHM COMPLETE
864	003324	000000	IDPTR:: .WORD	:PTR IN TESTID MACRO
865	003326	000000	TSTTOP:: .WORD	:POINTER TO CURRENT TEST
866	003330	000010	MXRTRY:: .WORD 8.	:NUMBER OF RETRIES, BEFORE 'HARD' ERROR
867	003332	000000	BLKER:: .WORD	:RECORD # FOR ERROR REPORT
868	003334	000000	SECREC:: .WORD	:RECORD # TO START AFTER SWITCHING TRACKS
869	003336	000000	PRNSIZ:: .WORD	:SIZE OF PACK FOR 'PRNPAK'
870	003340	000000	ALLGON:: .WORD	:SECONDARY DON'T PRINT STATISTICS FLAG.
871	003342	000000	TEST9:: .WORD	:***** FLAG THAT INDICATES IF IN TST 9
872				
873				
874				
875				
876				
877	003344	177777		:TIME OUT ON SEND DELAY (DLV NOT READY)
878	003346	000226	CSNRDY:: -1	:45 SEC. MULTIPLIER REWIND TIME
879			CSRCVB:: 150.	:CAN BE OPTIMISED PER CPU

TIME OUT CONSTANTS:

```

881      .SBTTL  DATA BLOCK FORMAT
882      -----
883      :R5 --> TOP OF 1 OF THE 8 DATA BLOCKS (1 PER UNIT) DURING EXECUTION
884      :@R5 IS THE STATUS WORD CONTAINING:
885      ;BIT15 = ABORTED
886      ;BIT14 = SEND 'BREAK'
887      ;BIT13 = RETRY FLAG BYTE ERROR (DATA PACKS)
888      ;BIT12 = TEMP STOR WRITE MACRO
889      ;BIT11 = UNIT NOT BEING TESTED
890      ;BIT10 = RETRYING DATA ERROR
891      ;BIT9  = TU58 CHKSUM ERROR
892      ;BIT8  = RD/WR OPERATION
893      ;BIT7  = NORMAL/REDUCED THRESHOLD (MACROS)
894      ;BIT6  = HOST DATA COMPARE ERROR
895      ;BIT5  = WR VERIFY OPERATION
896      ;BIT4  = TYPE OF PAK SENT (DATA 1CMD)
897      ;BIT3  = RETRY FLAG BYTE ERR. (SEND COMMAND PACK)
898      ;BIT0,1,2=UNIT NO.
899      ;DEVICE STATE
900      ;# OF RETRIES
901      ;ERROR NUMBER FOR LOG
902      ;STORAGE FOR REGISTERS USED IN TEST BODY
903      ;STORED WITH SWAPOW
904      ;RETRIEVED WITH SWAPIN
905      :
906      :
907      ;POINTER TO NEXT EXECUTABLE TEST INST.
908      ;DLV RCV STATUS ADDRESS
909      ;DLV RCV DATA ADDRESS
910      ;DLV SND STATUS ADDRESS
911      ;DLV SND DATA ADDRESS
912      ;THE NUMBER OF PACKETS TO RECEIVE
913      ;THE EXPECTED FLAG OF 1ST PACKET
914      ;THE EXPECTED COUNT OF 1ST PACKET
915      ;FOR MULTIPLE PACKET RECIEVES (MAX.4)
916      ;CONSECUTIVE XSFLGS AND XSCNTS
917      ;DR==0 OR 1; BIT8,9 DRIVE SELECTED BY OPERATOR
918      ;COUNTER FOR TRACK NUMBER
919      ;RECORD (BLOCK #)
920
921      ;TEST MACRO REGISTER
922      ;THE # OF BYTES FOR SENDING PACKET
923      ;DATA PATTERN-LOWER BYTE USED
924      ;CONTENTS OF RCDB ON DLV ERROR
925      ;SUCCESS CODE OF LAST END PACKET
926      ;TYPE OF COMMAND CURRENT IN EVEN BYTE; BIT15==VERIFY OP.
927
928      ;POINTER TO 542. BYTE BUFFER (4 DATA PAKS + END PACK)
929      ;POINTER TO TOP OF PACKET
930      ;POINTER TO CURRENTLY USED XSFLG OR XSCNT
931      ;THE # OF 512. BYTE BLOCKS WRITTEN DR0
932      ;THE # OF 512. BYTE BLOCKS WRITTEN DR1
933      ;THE # OF 512. BYTE BLOCKS READ DR0
934      ;THE # OF 512. BYTE BLOCKS READ DR1

```

```

936      ;AND THE ERROR LOG...      +-----+
937      ;SPLIT INTO A BYTE PER DRIVE:  ! DR1 ! DR0 !
938      ;                               +-----+
939
940      ;-----+-----+-----+-----+
941      ;OFFSET IN DATA BLOCK      ;ERROR TYPE      ;ERRCODE;MSG CODE;SUC. CODE
942      ;-----+-----+-----+-----+
943
944      000120      LGOFS      ==      80.      ;**RESERVED**
945      000122      SOFTR      ==      82.      ;SOFT READ      ;SFTRD      ;MSSFRD      ;ESCKSM
946      000124      SOFTW      ==      84.      ;SOFT WRITE      ;SFTWR      ;MSSFWR      ;ESSKSM
947      ;          WORD      ;RECIEVED INIT      ;RCINIT      ;MSRNIT      ;*****
948      ;          WORD      ;BAD FLAG BYTE      ;OTL      ;MSQRSP      ;*****
949
950      ;THEN THOSE CODES WHICH HAVE N TRIES BEFORE ABORT
951
952      000132      T4TRY      ==      90.      ;DLV ERROR      ;OVRN      ;MSOVRN      ;*****
953      000134      BDATA      ==      92.      ;BAD DATA      ;BDCOM      ;MSDATA      ;*****
954      000136      HARDR      ==      94.      ;HARD READ      ;HRDRD      ;MSHRD      ;ESCKSM
955      000140      HARDW      ==      96.      ;HARD WRITE      ;HRDWR      ;MSHDWR      ;ESCKSM
956      ;          WORD      ;CHKSM AT HOST      ;BDCHK      ;MSHCHK      ;*****
957      ;          WORD      ;SEEK ERROR TOTAL      ;SKERR      ;MSSKER      ;*****
958      000146      T1TRY      ==      102.      ;WRITE PROTECT      ;WRLOCK      ;MSWPRO      ;ESWLOC
959      ;          WORD      ;NO MOTOR      ;NOMOT      ;MSNOMO      ;ESNOMO
960      ;          WORD      ;CANT INIT      ;CNINIT      ;MSNIT      ;*****
961      ;          WORD      ;PARTIAL OP      ;PARTL      ;MSPART      ;ESPART
962      ;          WORD      ;NO UNIT      ;NOUNIT      ;MSUNIT      ;ESNONX
963      ;          WORD      ;COMMAND ERROR      ;CMNDER      ;MSCMD      ;ESCMD
964      ;          WORD      ;BAD RECORD NO.      ;RECERR      ;MSREC      ;ESREC
965      ;          WORD      ;SELF TEST ERROR      ;SLFER      ;MSSELF      ;*****
966      ;          WORD      ;WRONG SUC.CODE      ;SUCOTL      ;MSWRSP      ;*****
967      ;          WORD      ;NO RESPONSE      ;TORCVB      ;MSNRSP      ;*****
968      ;          WORD      ;**RESERVED**
969      ;          WORD      ;NO CARTRIDGE      ;NOCART      ;MSNOTP      ;ESNCRT
970      ;          WORD      ;TIME OUT SEND      ;TOSNDB      ;MSTOSN      ;*****
971
972      000202      BLKEND      ==      130.      ;OFFSET OF END OF STATISTICS (RESERVED)
973      000204      TUVECT      ==      132.      ;VECTOR ADDRESS
974      000206      SAVCNT      ==      134.      ;BYTE COUNT SAVED DURING RETRY ON WRITE OPERATION
975      000210      MRSP      ==      136.      ;***** FLAG INDICATING MRSP
976      000212      BLKSIZ      ==      138.      ;** RESERVED **
977      ;-----+-----+-----+-----+
978

```

```
981          .SBTTL  DEVICE DATA BLOCK ALLOCATION
982
983
984          ;TABLE OF DEVICE DATA BLOCK ADDRESSES
985
986          BLKTBL::      .WORD  DEV0
987 003350    003370      .WORD  DEV1
988 003352    003602      .WORD  DEV2
989 003354    004014      .WORD  DEV3
990 003356    004226      .WORD  DEV4
991 003360    004440      .WORD  DEV5
992 003362    004652      .WORD  DEV6
993 003364    005064      .WORD  DEV7
994 003366    005276
995
996          LSTDEV::
997
998          ;AND STORAGE FOR EACH:
999
1000 003370    DEV0:      .BLKB  BLKS1Z
1001 003602    DEV1:      .BLKB  BLKS1Z
1002 004014    DEV2:      .BLKB  BLKS1Z
1003 004226    DEV3:      .BLKB  BLKS1Z
1004 004440    DEV4:      .BLKB  BLKS1Z
1005 004652    DEV5:      .BLKB  BLKS1Z
1006 005064    DEV6:      .BLKB  BLKS1Z
1007 005276    DEV7:      .BLKB  BLKS1Z
```

1022 .SBTTL GLOBAL TEXT SECTION
1023
1024
1025 : NAMES OF DEVICES SUPPORTED BY PROGRAM
1026 :
1027 : DEVTYP <TU58 CONTROLLER>
005510
005510
005510 124 125 065

1028
1040
1041
1059

L\$DVTYP::
.ASCIZ /TU58 CONTROLLER/
.EVEN

```

1068      .SBTTL  SYSTEM MACRO DEFINITIONS
1069
1070      .MACRO  PUSH      ,REG
1071
1072      .NLIST
1073      .LIST ME
1074      .LIST
1075                      MOV      REG,-(SP)
1076
1077
1078      .NLIST
1079      .NLIST ME
1080      .LIST
1081      .ENDM
1082
1083      .MACRO  POP,REG
1084
1085      .NLIST
1086      .LIST ME
1087      .LIST
1088                      MOV      (SP)+,REG
1089
1090
1091      .NLIST
1092      .NLIST ME
1093      .LIST
1094      .ENDM
1095
1096      ;++
1097      ;THE MACRO 'SWAPIN' RETRIEVES THE TEST REGISTERS WHICH WERE SAVED
1098      ;IN THE DEVICE DATA BLOCK.
1099      ;--
1100
1101      .MACRO  SWAPIN
1102
1103      .NLIST
1104      .LIST ME
1105      .LIST
1106                      MOV      6.(R5),R0
1107                      MOV      8.(R5),R1
1108                      MOV      10.(R5),R2
1109                      MOV      12.(R5),R3
1110                      MOV      14.(R5),R4
1111
1112      .NLIST
1113      .NLIST ME
1114      .LIST
1115      .ENDM
1116
1117      ;++
1118      ;THE MACRO 'SWAPOW' SAVES THE CURRENT STATE OF THE UNIT IN THE DRIVE
1119      ;DATA BLOCK IN SO THAT THE SCHEDULER MAY 'SWAPIN' ANOTHER UNIT.
1120      ;--
1121
1122      .MACRO  SWAPOW
1123
1124      .NLIST
1125      .LIST ME

```


1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135

.LIST

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

.NLIST
.NLIST ME
.LIST
.ENDM

```
1138      :++
1139      :THE WRITE MACRO IMPLEMENTS THE COMPLETE PROTOCOL NECESSARY TO BUILD
1140      :A COMMAND PACKET AND SUBSEQUENT DATA PACKETS (UNTIL THE BYTE COUNT
1141      : (BCNT) IS SATISFIED).
1142      :
1143      :SETS UP THE EXPECTED PROTOCOL RESPONSES: THE NUMBER OF PACKETS
1144      : (XSPKMN) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT). CALLS
1145      : 'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
1146      : CHECKSUM.
1147      :
1148      : INPUTS - DEVICE BLOCK @R5
1149      :          TRBUF - BUFFER ADDRESS
1150      :          UNIT'S TEST REGISTERS FROM 'SWAPIN'
1151      : OUTPUTS - SND CNT(R5) = # OF BYTES TO SEND
1152      :          XSPKMN = # OF PACKETS EXPECTED
1153      :          XSFLG = FLAG BYTE OF 1ST PACKET
1154      :          XSCNT = BYTE COUNT OF 1ST PACKET
1155      :          ***
1156      :          * SUBSEQUENT XSFLGS
1157      :          * >
1158      :          * AND XSCNTS
1159      :          ***
1160      :--
1161
1162      .MACRO TUWRIT PTRN,REC,BCNT,DR,VER,?A,?B,?C,?D,?E,?F,?G,?H,?T
1163
1164      .NLIST
1165      .LIST ME
1166      .LIST
1167
1168      T:      MOV      #TRBUF,R0      ;MAKE COMMAND PACKET:
1169      MOV      #RSCMND,@R0      ;COMMAND FLAG
1170      MOV      #RSMSIZ,1(R0)      ;THIS SIZE
1171      MOV      #RSSWR,2(R0)      ;INSERT OP CODE-WRITE
1172      MOV      VER,3.(R0)      ;VERIFY (1 OR 0)
1173      MOV      DR,4.(R0)      ;DRIVE #
1174      MOV      #020,5.(R0)      ;MAINTENANCE MODE SWITCH
1175      CLR      6.(R0)      ;NO SEQUENCE #
1176      MOV      BCNT,8.(R0)      ;TOTAL COUNT TO WRITE
1177      MOV      REC,10.(R0)      ;AT RECORD N
1178      MOV      #RSMSIZ,R1      ;THE PACKET SIZE PLUS+2
1179      TST      (R1)+      ;((FLAG AND COUNT) INTO R1
1180      MOV      #RSSNSZ,SND CNT(R5) ;LOAD THE SIZE TO SEND
1181      CALL     CHKSUM      ;R0 --> R1=COUNT
1182      MOV      R1,(R0)      ;PUT CHKSUM IN PACKET
1183      ;SET UP EXPECTATIONS:
1184      MOV      #RSCNT,XSFLG(R5) ;THE FLAG
1185      MOV      #1,XSCNT(R5)      ;THE COUNT
1186      MOV      #1,XSPKMN(R5)      ;THE # PACKETS EXPECTED
1187      MOV      BCNT,R2      ;GET # OF DATA BYTES
1188      CALL     RSVP      ;SEND (AND RETURN TO SCHEDULER)
1189      BIT      #BIT3,@R5      ;FLAG BYTE ERROR?
1190      BNE      T      ;YES
1191      BIC      #BIT12,@R5      ;FLAG FOR LAST PACKET
1192      MOV      #TRBUF,R0      ;POINT TO TOP OF BUFFER AGAIN
1193      CMP      R2,#128.      ;START DATA PACKET(S)
1194      BHI      B      ;BCNT > 128.!
1195      MOV      R2,R1      ;BCNT<128.
```



```
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
```

```

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

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

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
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301

++
THE SEEK MACRO IMPLMENTS THE COMPLETE PROTOCOL TO INITIATE A SEEK
SEQUENCE.

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

INPUTS - DEVICE BLOCK @RS
UNITS TEST REGISTERS FROM SWAPIN
TRBUF - BUFFER ADDRESS

OUTPUTS -
XSPKMN = # 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
	MOVB	#RSCMND,@R0	:FORM COMMAND MESSAGE PACK
	MOVB	#RSMSIZ,1(R0)	:THIS BIG
	MOVB	#RSSEK,2(R0)	:OP CODE IS SEEK
	MOV	REC,10.(R0)	:TO THIS RECORD
	MOVB	DR,4.(R0)	:AND WHICH DRIVE
	CLRB	3.(R0)	:NO MODIFIER
	CLRB	5.(R0)	:NO SWITCHES
	CLR	6.(R0)	:NO SEQUENCE #
	CLR	8.(R0)	:NO BYTE COUNT
	MOV	#RSMSIZ,R1	:GET COUNT
	TST	(R1)+	:PLUS FLAG + BCNT
			:FOR CHECKSUM CALC
	CALL	CHKSUM	:RO-->TOP R1=# OF BYTES
	MOV	R1,(R0)	:INSERT INTO PACKET
			:SET UP EXPECTATIONS:
	MOV	#RSSNSZ,SNDcnt(R5)	:HOW MANY TO SEND
	MOVB	#RSCMND,XSFLG(R5)	:EXPECT END PACK
	MOV	#RSNDSZ,XSCNT(R5)	:COUNT WITH THIS
	MOV	#1,XSPKMN(R5)	:EXPECT ONLY 1 PACKET
	CALL	RSVP	:SEND
			:AND RETURN TO SCHEDULER
	BIT	#BIT3,@RS	:RETRY (FLAG BYTE ERROR)?
	BNE	A	:YES

1302
1303
1304
1305

.NLIST
.NLIST ME
.LIST
.ENDM

```

1308      ;++
1309      ;THE RETRY MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO INITIATE
1310      ;A RETRY (READ OPERATION) SEQUENCE.
1311      ;
1312      ;SETS UP THE EXPECTED PROTOCOL RESPONSES: THE NUMBER OF PACKETS
1313      ;(XSPKMM) AND THEIR FLAG BYTES AND COUNTS (XSFLG, XSCNT). CALLS
1314      ;'RSVP' TO SEND EACH PACKET, AND 'CHKSUM' TO CALC. THE PACKET
1315      ;CHECKSUM.
1316      ;
1317      ; INPUTS - DEVICE BLOCK @R5
1318      ;          TRBUF - BUFFER ADDRESS
1319      ;          UNITS TEST REGISTERS FROM SWAPIN
1320      ;
1321      ; OUTPUTS - SND CNT(R5) = # OF BYTES TO SEND
1322      ;           XSPKMM = # OF PACKETS EXPECTED
1323      ;           XSFLG = FLAG BYTE OF 1ST PACKET
1324      ;           XSCNT = BYTE COUNT OF 1ST PACKET
1325      ;           ***
1326      ;           * SUBSEQUENT XSFLGS
1327      ;           * >
1328      ;           * AND XSCNTS
1329      ;           ***
1330      ;--
1331
1332      .MACRO TURTRY REC,BCNT,DR,?A,?B,?C,?D,?E
1333
1334      .NLIST
1335      .LIST ME
1336      .LIST
1337
1338      D:      MOV      #TRBUF,R0      ;FORM CMD PACK:
1339      MOV      #RSCMD,@R0      ;MESSAGE PACK TYPE
1340      MOV      #RSMSIZ,1(R0)      ;THIS BIG
1341      MOV      #RSSRD,2(R0)      ;OP CODE-READ
1342      MOV      REC,10.(R0)      ;THIS RECORD
1343      MOV      DR,4.(R0)      ;THIS DRIVE
1344      CLRB     3(R0)      ;PRESET NORM THRESHOLD
1345      TSTB     @R5      ;REDUCED?
1346      BPL      E      ;NO
1347      INCB     3(R0)      ;YES-CHANGE THRESHOLD
1348      E:      MOV      BCNT,8.(R0)      ;# BYTES DESIRED
1349      MOV      #020,5.(R0)      ;MAINTENANCE MODE
1350      CLR      6.(R0)      ;NO SEQUENCE #
1351      MOV      #RSMSIZ,R1      ;SIZE OF PACKET
1352      TST      (R1)+      ;PLUS FLAG+COUNT INTO R1
1353      MOV      #RSSNSZ,SND CNT(R5) ;SET UP SIZE TO SEND
1354
1355      CALL     CHKSUM      ;FORM CHECKSUM R1=COUNT
1356      MOV      R1,(R0)      ;INSERT IN PACKET
1357
1358      MOV      BCNT,R1      ;SET EXPECTATIONS:
1359
1360      MOV      #XSFLG,R3      ;CALC # OF DATA PACKETS TO EXPECT
1361      ADD      R5,R3      ;OFFSET OF FLAG
1362      CLR      R2      ;ABS. ADDR. OF XSFLG
1363      INC      R2      ;PRESET
1364      A:      INC      R2      ;# PACKETS EXPECTED
1365      MOV      #RSDATA,(R3)+ ;LOAD XSFLG

```

1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380

```
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      #PSEND,(R3)+   ;EXPECT AN END
        MOV      #RSNDSZ,(R3)   ;THIS BIG-14. BYTES

        CALL     RSVP           ;SEND
                                   ;AND RETURN TO SCHEDULER
```

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

1383
1384
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
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439

;;
;THE READ MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO INITIATE
;A READ SEQUENCE.

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

INPUTS - DEVICE BLOCK @R5
TRBUF - BUFFER ADDRESS
UNITS TEST REGISTERS FROM SWAPI.N

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

. ***
. * SUBSEQUENT XSFLGS
. *
. * AND XSCNTS
. ***

--

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

.NLIST
.LIST ME
.LIST

E:	MOV	#TRBUF,R0	;FORM CMND PACK:
	MOVB	#RSCMND,@R0	;MESSAGE PACK TYPE
	MOVB	#RSMSIZ,1(R0)	;THIS BIG
	MOVB	#RSSRD,2(R0)	;OP CODE IS READ
	MOV	REC,10.(R0)	;THIS RECORD
	MOVB	DR,4.(R0)	;THIS DRIVE
	MOVB	VER,3.(R0)	;VERIFY
	MOV	BCNT,8.(R0)	;TOTAL BYTES TO READ
	MOVB	#O20,5.(R0)	;MAINTENANCE MODE
	CLR	5.(R0)	;NO SEQUENCE #
	MOV	#RSMSIZ,R1	;GET SIZE OF PACKET
	TST	(R1)+	;+2 FOR CHECKSUM
	MOV	#RSSNSZ,SND CNT(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:
	MOV	#XSFLG,R3	;GET OFFSET
	ADD	R5,R3	;ABS. ADDR. OF XSFLG
	CLR	R2	;PRESET AS NONE
A:	INC	R2	;# PACKETS EXPECTED
	MOV	#RSDATA,(R3)+	;LOAD XSFLG
	MOV	#132.,(R3)+	;AND EXPECTED COUNT
	SUB	#128.,R1	;NEG RESULT LAST TIME
	BLOS	C	;LAST TIME

1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457

C:	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 ALSO...
	MOV	#RCNDSZ,(R3)	;THIS BIG-14. BYTES
	CALL	RSVP	;SEND
			;AND RETURN TO SCHEDULER
D:	BIT	#BIT10!BIT3,DR5	;RETRY?
	BEQ	B	;NO.
	TURTRY	REC,BCNT,DR	;YES
	BR	D	;ANOTHER RETRY?
B:	NOP		;NO

.NLIST
.NLIST ME
.LIST
.ENDM

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
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515

```

:++
:THE SELF TEST MACRO IMPLIMENTS THE COMPLETE PROTOCOL NECESSARY TO
:INITIATE A 'DIAGNOSE' SEQUENCE.

```

```

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

```

```

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

```

```

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

```

.MACRO TUSELF ?A

.NLIST
.LIST ME
.LIST

```

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

```

.NLIST
.NLIST ME
.LIST
.ENDM


```

1518      :++
1519      :THE TEST ID MACRO INTERFACES THE SUPERVISOR'S TEST DISPATCH TO THE
1520      :DIAGNOSTIC'S FORMAT BY IMPLEMENTING CALLS THAT: 1) INITIALIZE THE
1521      :PC OF THE TEST CODE (TSTPC(R5)), 2) ASSIGN THE 1ST DRIVES, 3) RUN
1522      :THE TEST, 4) SWITCH DRIVES AND REINITIALIZE, 5) RUN THE TEST AGAIN.
1523      :--
1524
1525      .MACRO TSTID  ADDRESS,?A
1526
1527      .NLIST
1528      .LIST ME
1529      .LIST
1530
1531      MOV      ADDRESS,TSTTOP      ;SAVE ADDR OF TEST
1532      CALL     SETUP              ;INIT UNITS TSTPC
1533      CALL     SETDR              ;GET 1ST DRVS.
1534      CALL     RUN                ;DO TEST
1535      CALL     SWAPDR             ;GET NEXT DRVS.
1536      BCC      A                 ;BR NO 2ND DRVS
1537      CALL     SETUP              ;REINIT UNITS TSTPC
1538      CALL     RUN                ;REPEAT TEST
1539
1540      A:
1541      .NLIST
1542      .NLIST ME
1543      .LIST
1544      .ENDM
1545
1546      ;-----

```

```

1546      .SBTTL  GLOBAL SUBROUTINES SECTION
1547
1548      :++
1549      : THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES THAT ARE USED
1550      : TO LINK THE DIAGNOSTIC TO THE SUPERVISOR (THROUGH THE TSTID MACRO).
1551      :--
1552
1553      :++
1554      : SWAPDR
1555      : SUBROUTINE TO DETERMINE IF TO TEST OTHER DRIVE (FOR ALL UNITS)
1562      : INPUTS:  DR(R5) - DRIVE CONFIGURATION
1563                  BLKTBK - TOP OF DATA BLOCK ALLOCATION TABLE
1564                  LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
1565
1566      : OUTPUTS:  DR(R5) UPDATED TO TEST SAME OR OTHER DRIVE
1567                  CARRY SET IF SECOND PASS NECESSARY
1610      :--
1622
1623 005530 005002 SWAPDR:: CLR R2 ;FOR # OF DRIVE 1'S.
1624 005532 012737 003350 005630 MOV #BLKTBK,SWPTR ;TABLE ADDR. OF 1ST UNIT
1625 005540 017705 000064 1$: MOV @SWPTR,R5 ;GET DATA BLOCK ADDR.
1626 005544 032715 100000 BIT #BIT15,@R5 ;ABORTED?
1627 005550 001013 BNE 3$ ;YES
1628 005552 032765 000001 000060 BIT #BIT0,DR(R5) ;DID DR. 0?
1629 005560 001007 BNE 3$ ;NO, DID DR.1 1ST PASS
1630 005562 032765 001000 000060 BIT #BIT9,DR(R5) ;YES; 1 SELECTED?
1631 005570 001403 BEQ 3$ ;NO, ALL DONE
1632 005572 105265 000060 INCB DR(R5) ;YES, SWAP
1633 005576 005202 INC R2 ;ONE MORE TO TEST
1634 005600 023727 005630 003366 3$: CMP SWPTR,#LSTDEV ;LAST DEVICE?
1635 005606 103004 BHIS 4$ ;YES
1636 005610 062737 000002 005630 ADD #2,SWPTR ;NO-POINT NEXT
1637 005616 000750 BR 1$ ;DO
1638
1639 005620 005702 4$: TST R2 ; (CLEAR CARRY), MORE TO DO?
1640 005622 001401 BEQ 5$ ;NO
1641 005624 000261 SEC ;YES
1642 005626 000207 5$: RETURN ;RETURN
1643
1644 005630 000000 SWPTR: .WORD

```

```

1647
1648      :++
1649      : SETDR - SUBROUTINE TO GET DRIVE FOR 1ST PASS FOR EACH TEST
1650      :
1651      : INPUTS:      DR(R5) - DRIVE CONFIGURATION
1652      :              BLKTB1 - TOP OF DATA BLOCK ALLOCATION TABLE
1653      :              LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
1654      :
1655      : OUTPUTS:     DR(R5) IS SET TO TEST DRIVE 0 OR DRIVE 1
1656      :
1657
1658 005532 012737 003350 005706 SETDR:: MOV      #BLKTB1,SETPTR ;TABLE OF ADDR. 1ST UNIT
1659 005540 017705 000042 1$:      MOV      @SETPTR,R5 ;GET DATA BLOCK ADDR.
1660 005544 105065 000060      CLR      DR(R5) ;PRESET AS DRO
1661 005650 032765 000400 000060      BIT      #BIT8,DR(R5) ;DO DRO?
1662 005556 001002      BNE      2$ ;YES
1663 005660 105265 000060      INCB     DR(R5) ;NO-USE DRIVE 1
1664 005564 023727 005706 003366 2$:      CMP      SETPTR,#LSTDEV ;MORE UNITS
1665 005 72 103004      BHS      7$ ;NO-EXIT
1666 005674 062737 000002 005706      ADD      #2,SETPTR ;YES-GET TABLE ENTRY
1667 005702 000756      BR       1$ ;CONFIGURE THAT UNIT
1668 005704 000207      3$:      RETURN
1669 005706 000000      SETPTR: .WORD

```

```

1672      ;**
1673      ; CLRALL - CLEARS INPUT BUFFER FOR RESPONSE FROM UNIT.
1674      ;
1675      ; INPUTS:      BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
1676      ;              LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
1677      ;
1678      ; OUTPUTS:     ALL UNITS BUFFERS CLEARED.
1679      ;
1680      ; CALLS:       CLRBUF
1681      ;--
1682
1683 005710 012737 003350 006002 CLRALL:: MOV  #BLKTBL,CLRPTR ;TOP OF TABLE OF ADDRESSES
1684 005716 017705 000060 1$:      MOV  @CLRPTR,R5 ;GET DATA BLOCK
1685 005722 004737 005750      CALL CLRBUF ;CLEAR IT'S RECEIVE BUFFER
1686 005726 023727 006002 003366      CMP  CLRPTR,#LSTDEV ;LAST DEV?
1687 005734 103004      BHS  2$ ;YES
1688 005736 062737 000002 006002      ADD  #2,CLRPTR ;-->NEXT
1689 005744 000764      BR    1$ ;CONTINUE
1690 005746 000207      2$:      RETURN

```

```

1693
1694
1695
1696
1697
1698
1699
1700
1701 005750 010046 CLRBUF:: PUSH R0 ;SAVE R0
;MOV R0,-(SP)

1702 005752 010446 PUSH R4 ;SAVE R4
;MOV R4,-(SP)

1703 005754 016500 000102 MOV RCVBUF(R5),R0 ;GET ADDRESS OF BUFFER
1704 005760 012704 001036 MOV #RCBFSZ,R4 ;SIZE IN BYTES
1705 005764 005020 1$: CLR (R0)+ ;CLEAR IT
1706 005766 162704 000002 SUB #2,R4 ;2 BYTES LESS
1707 005772 001374 BNE 1$ ;MORE
1708 005774 012604 POP R4 ;RESTORE
;MOV (SP)+,R4

1709 005776 POP R0 ;
;MOV (SP)+,R0

1710 006000 000207 RETURN ;EXIT
1711 006002 000000 CLRPTR: .WORD

```

```

1714
1715      :++
1716      : SETUP - CALLED WITHIN EACH TEST TO INSERT BEGINNING ADDRESS OF THE
1717      : TEST INTO ALL UNITS TEST PC'S.
1718      : INPUTS:  TSTTOP LOADED WITH TEST ALGORITHMS STARTING ADDR.
1719      :           BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
1720      :           LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
1721      :
1722      : OUTPUTS:  TSTPC(R5) FOR ALL UNITS
1723      :           DONE - CLEARED
1724      :--
1725 006004 005037 003322  SETUP:: CLR      DONE      :NOT DONE YET
1726 006010 012737 003350 003324  MOV      #BLKTBL, IDPTR :TABLE TOP ADDR
1727 006016 017705 175302      1$:  MOV      @IDPTR, R5    :DEVICE'S DATA BLOCK
1728 006022 013765 003326 000020      MOV      TSTTOP, TSTPC(R5) :INSERT PC FOR TOP OF TEST
1729 006030 023727 003324 003366      CMP      IDPTR, #LSTDEV :ALL UNITS SET?
1730 006036 103004      BHIS      2$      :YES
1731 006040 062737 000002 003324      ADD      #2, IDPTR    :NO, GET NEXT POINTER
1732 006046 000763      BR        1$      :SET HIM UP
1733 006050 000207      2$:  RETURN    :DONE

```



```

1736
1737
1738
1739
1740
1741
1742
1743 006052 004737 006102
1744
1745 006056 005737 003322
1746 006062 001006
1747 006064 004737 007152
1748
1749 006070
      006070 104422
1750
1751 006072 004737 010576
1752 006076 000765
1753 006100 000207

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

RUN::  CALL  NXTST      ;MAKE AND SEND NEXT PACK TO ALL
      TST    DONE      ;UNABORTED UNITS
      BNE    2$         ;COMPLETE?
      CALL   GETANS     ;YES
                        ;NO,GET ALL RESPONSES

      BREAK           ;SUPERVISOR CHECK
                        TRAP    C$BRK

      CALL    CHKANS    ;CHECK ALL RESPONSES
      BR      RUN       ;CONTINUE TILL DONE

2$:    RETURN

```

```
1756 .SBTTL NXTST / THE SCHEDULER
1757
1758 :++
1759 : NXTST - DISPATCH EXECUTION USING EACH UN-ABORTED UNIT'S TEST PROGRAM
1760 : COUNTER, (TSTPC(R5)). (THE POINTER TO THE TEST CODE THAT COMPRISES
1761 : MAKING A PACKET AND SENDING IT. CHECKS FIRST FOR ANY UN-ABORTED UNIT
1762 : THAT IS RETRYING EITHER A DATA ERROR OR A 'INDECIPHERABLE FLAG BYTE'
1763 : ERROR, IN ORDER TO SERVICE ONLY THAT UNIT THIS PASS. INITIS
1764 : NON-RETRYING UNITS IF NECESSARY. IF NO RETRIES, DISPATCH ALL
1765 : UNITS IN ROUND ROBIN FASHION.
1766
1767 : INPUTS: (IMPLIED) DATA BLOCKS.
1768 : BLKTBL - TOP OF DATA BLOCK ALLOCATION TABLE
1769 : LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
1770
1771 : OUTPUTS: ERRSF IF ALL UNITS ARE ABORTED. (TO NOTIFY APT)
1772 : SYSTAT IS UPDATED
1773 :--
1774
1775 006102 000240 NXTST:: NOP
1776 006104 012737 003350 003312 MOV #BLKTBL,DEVPTR ;UNIT 0 TO START
1777 006112 017705 175174 1$: MOV @DEVPTR,R5 ;GET DATA BLOCK
1778 006116 005715 TST @R5 ;ABORTED?
1779 006120 100504 BMI 2$ ;YES... CHECK NEXT UNIT
1780 006122 032715 000010 3$: BIT #BIT3,@R5 ;NO-RETRY 'B' FLAG?
1781 006126 001040 BNE 5$ ;YES...(SEND ACK; THEN CMD PACK)
1782 006130 032715 020000 BIT #BIT13,@R5 ;NO-RETRYING STILL (NO END PACK YET)?
1783 006134 001426 BEQ 7$ ;NO...
1784 006136 032715 000400 BIT #BIT8,@R5 ;RETRYING A WRITE?
1785 006142 001453 BEQ 4$ ;NO...
1786 006144 SWAPIN ;YES-GET DEVICE REGESTERS
1787 006170 020265 000206 CMP R2,SAVCNT(R5) ;CURRENT COUNT = SAVED COUNT? (WHERE WE STARTED)
1788 006174 001036 BNE 4$ ;NO...(CONTINUE SENDING DATA PACKS)
1789 006176 042737 000004 003306 BIC #BIT2,SYSTAT ;YES-CLEAR RETRY FLAGS
1790 006204 042715 020000 RIC #BIT13,@R5
1791 006210 000450 BR 2$ ;CHECK NEXT UNIT.
1792 006212 032715 002000 7$: BIT #BIT10,@R5 ;NO-RETRY DATA ERROR?
1793 006216 001445 BEQ 2$ ;NO...ON TO NEXT UNIT
1794 006220 052737 000002 003306 BIS #BIT1,SYSTAT ;SET RETRY STATUS TO 'DATA ERROR' TYPE
1795 006226 000424 BR 6$ ;YES...
1796
1797 006230 5$: SWAPIN ;GET DEVICE REGISTERS
1798 006230 016500 000006 MOV 6.(R5),R0
1799 006234 016501 000010 MOV 8.(R5),R1
1800 006240 016502 000012 MOV 10.(R5),R2
006244 016503 000014 MOV 12.(R5),R3
006250 016504 000016 MOV 14.(R5),R4
1798 006254 010265 000206 MOV R2,SAVCNT(R5) ;SAVE THE BYTE COUNT (FOR WRITE OPERATION)
1799
1800 006260 004737 014010 CALL DOBRK ;TO MARK HOW MANY DATA PACKS TO SEND
;SEND INIT
```



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

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

1826 006444 004775 000020          JSR     PC,@TSTPC(R5)    ;INITIATE 1 PACKET TRANSMISSION AND RETURN
1827 006450 005715          4$:    TST     @R5              ;ABORTED?
1828 006452 100002          BPL     8$              ;NO-ON TO NEXT UNIT
1829 006454 005237 006532          INC     ABONM        ;YES...ONE MORE TALLIED
1830 006460 023727 003312 003366 8$:    CMP     DEVPTR,#LSTDEV ;ALL TU S TRIED?
1831 006466 103004          BHIS    5$              ;YES
1832 006470 062737 000002 003312          ADD     #2.,DEVPTR    ;NO THE ADDRESS+2=NEXT ADDRESS
1833 006476 000732          BR      PERDEV        ;DO NEXT UNIT
1834 006500 022737 000010 006532 5$:    CMP     #8.,ABONM    ;ALL ABORTED?
1835 006506 001010          BNE     NXTRET        ;NO
1836 006510          ERRSF    100.,NOMOR    ;YES!
      006510 104454          TRAP     CSERSF
      006512 000144          .WORD    100
      006514 006534          .WORD    NOMOR
      006516 000000          .WORD    0
1837 006520          11$:    BREAK          ;SUPERVISOR BREAK
      006520 104422          TRAP     CSBRK
1838 006522 005237 003340          INC     ALLGON        ;SET DON'T-PRINT STATISTICS FLAG
1839 006526          DOCLN          ;EXIT
      006526 104444          TRAP     CSDCLN
```

1840 006530 000207

NXTRET: RETURN

1841

1842 006532 000000

1843 006534 101

114

114

ABONM: .WORD

NOMOR: .ASCIZ

.EVEN

:THE NUMBER OF ABORTED UNITS
/ALL UNITS ABORTED!/
/

```

1846 .SBTTL RSVP / XOFF AND SEND A PACKET TO ALL DEVICES
1847
1848 :++
1849 : RSVP - SAVES TEST CODE PROGRAM COUNTER IN TSTPC(R5) AND UNIT'S REGIS-
1850 : TERS. IF NOT IN TEST 8, POINTS TO 'XOFF' THAT PRECEEDS PACKET IN
1851 : XMIT BUFFER AND SENDS PACKET WITH XOFF. RETURNS TO SCHEDULER (NXTST)
1852 : SO THAT OTHER UNITS PACKETS MAY BE FORMED, TO GET ALL UNITS WORKING
1853 : AT ONCE. IF IN TEST 8 AND THE UNIT IS NOT MODIFIED, SKIP REST OF
1854 : ROUTINE. IF IN TEST 8 AND THE UNIT IS MODIFIED DO NOT SEND XOFF AND
1855 : PROCEED NORMALLY.
1856 : INPUTS: (SP) CONTAINS UNITS PC TO SAVE SINCE RSVP WAS CALLED. THE
1857 : NUMBER PACKETS EXPECTED (XSPKNM), AND THE EXPECTED FLAGS AND
1858 : BYTE COUNTS OF EACH (XSFLG, XSCNT...) ARE LOADED BY TEST CODE
1859 : (MACROS).
1860 : SDCNT - # BYTES TO SEND
1861 : REC(R5) - RECORD #
1862 : TRBUF - BUFFER ADDR.
1863 : XSPKNM(R5) - # EXPECTED
1864 : RCVBUF(R5)
1865
1866 : OUTPUTS: CMDSNT - UPDATED WITH PACKET OP CODE
1867 : BLKER - RECORD NUMBER STATISTICS UPDATED IF NOT RETRYING
1868 : AND COMMAND PACKET SENT.
1869 : SUCCS(R5) - PRESET CLEAR
1870 : STATUS WORD @R5 - BIT9 - DATA CHECK ERROR - CLEARED
1871 : BIT5 - 'VERIFY' OPERATION
1872 : BIT4 - 0 = DATA PACK 1 = CMND
1873 : BIT8 - RD/WR OPERATION
1874 : XSPTR - POINTS TO EXPECTED FLAG
1875 : UPPER BYTE OF XSPKNM IS REPLICATED.
1876 : PACKET POINTER (PKPTR(R5)) POINTS TO TOP OF UNITS RECEIVE BUFFER
1877 : AREA (RCVBUF(R5)) FOR CURRENT UNIT.
1878 :--
1879
1880 RSVP:: NOP ;FINISH TEST
1881 MOV (SP)+,TSTPC(R5) ;SAVE WHERE YOU WERE IN TEST BODY AND
1882 SWAPOW ;SAVE TEST REGISTERS
1883
1884 MOV R0,6.(R5)
1885 MOV R1,8.(R5)
1886 MOV R2,10.(R5)
1887 MOV R3,12.(R5)
1888 MOV R4,14.(R5)
1889
1890
1891
1892
1893
1894
1895
1896
1883
1884
1885 006612 022737 000002 003342 CMP #2,TEST9 ;CORRECT FOR RETURN TO SCHEDULER
1886 006620 001007 BNE XFNSND ;***** IS THIS TEST 9
1887 006622 022765 000000 000210 CMP #0,MRSP(R5) ;***** NO
1888 006630 001523 BEQ ENDRSP ;***** IF SO, IS THIS UNIT MODIFIED
1889 006632 012700 027722 NOXOFF: MOV #TRBUF,R0 ;***** YES
1890 006636 000404 BR SND ;FOR NORMAL PACKET SEND
1891 006640 012700 027721 XFNSND: MOV #TRBUF-1,R0 ;SEND XOFF+PACKET
1892 006644 005265 000070 INC SDCNT(R5) ;POINT TO XOFF
1893 006650 004737 007102 SND: CALL SDCNT(R5) ;ONE MORE TO SEND, TOO.
1894 006654 005715 TST @R5 ;SEND BYTE
1895 006656 100510 BMI 6$ ;R5--> TO STATUS BLK
1896 006660 005365 000070 DEC SDCNT(R5) ;ABORTED? YES...QUIT
;NO, SEND MORE

```

```
1897 006664 001371          BNE      SND          ;IF MORE TO SEND
1898 006666 012700 027722    MOV      #TRBUF,R0        ;-->BUFFER
1899 006672 016537 000064 003332 MOV      REC(R5),BLKER    ;PREPARE FOR RECEIVE
1900 006700 156565 000032 000033 BISB    XSPKNN(R5),XSPKNN+1(R5) ;REPLICATE LO. BYTE TO HI FOR GTPAKS, CHKANS
1901 006706 005065 000076          CLR      SUCCS(R5)        ;NO SUCCESS YET
1902 006712 042715 001000          BIC      #BIT9,R5        ;NO DATA CHK ERROR YET
1903 006716 016565 000102 000104 MOV      RCVBUF(R5),PKPTR(R5) ;TOP OF RCV BUFFER GOES THE 1ST PACKET
1904 006724 012704 000034          MOV      #XSFLG,R4        ;FORM
1905 006730 060504          ADD      R5,R4          ;ADDRESS
1906 006732 010465 000106          MOV      R4,XSPTR(R5)      ;OF 1ST XSFLG
1907
1908 006736 042715 000020          BIC      #BIT4,R5        ;PRESET AS DATA PAK
1909 006742 121027 000002          CMPB    @R0,#RSCMND      ;WAS IT COMMAND PAK?
1910 006746 001054          BNE      6$              ;NO...
1911 006750 116065 000002 000100 MOVB    2(R0),CMDSNT(R5)    ;YES-SAVE COMMAND
1912 006756 052715 000020          BIS      #BIT4,R5        ;ITS CMND PAK
1913
1914 006762 032715 002000          BIT      #BIT10,R5       ;RETRYING?
1915 006766 001044          BNE      6$              ;YES-DON'T UPDATE ANY STATS OR CONDITION
1916 006770 126027 000002 000002 CMPB    2(R0),#RSSRD      ;NO,A READ?
1917 006776 001012          BNE      4$              ;NO
1918 007000 042715 000400          BIC      #BIT8,R5        ;(FOR HARD/SOFT LOGGING) RD/WR FLAG=0
1919 007004 004737 01364C          CALL    WHCHDR        ;GET DRIVE
1920 007010 103403          JCS      8$              ;
1921 007012 005265 000114          INC      RDN0(R5)        ;DRIVE 0
1922 007016 000402          BR       4$              ;
1923 007020 005265 000116          8$: INC      RDN1(R5)        ;DRIVE 1
1924
1925 007024 126027 000002 000003 4$: CMPB    2(R0),#RSSWR      ;A WRITE?
1926 007032 001022          BNE      6$              ;NO
1927 007034 052715 000400          BIS      #BIT8,R5        ;YES, RD/WR FLAG=1
1928 007040 105760 000003          TSTB    3(R0)          ;VERIFY TOO?
1929 007044 001403          BEQ      21$          ;NO
1930 007046 052715 000040          BIS      #BIT5,R5        ;YES-SET VERIFY FLAG
1931 007052 000402          BR       22$          ;
1932 007054 042715 000040          21$: BIC      #BIT5,R5        ;(NO)-RESET VERIFY FLAG
1933 007060 004737 013640          22$: CALL    WHCHDR        ;GET DRIVE NO
1934 007064 103403          BCS      5$              ;CARRY=DR1
1935 007066 005265 000110          INC      WRTN0(R5)      ;# BLKS WRITTEN DRO
1936 007072 000402          BR       6$              ;EXIT
1937
1938 007074 005265 000112          5$: INC      WRTN1(R5)      ;# BLKS WRITTEN DRV1
1939 007100          6$:
1940 007100 000207          ENDRSP: RETURN      ;RETURN
```

```

1943 .SBTTL SNDBYT / OUTPUT A BYTE TO UNIT
1944
1945 :++
1946 : SNDBYT - TEST 'READY' ON INTERFACE. IF 'READY', SEND BYTE AND EXIT.
1947 : IF TIMED OUT, LOG ERROR.
1948 : INPUTS - R0 = POINTER TO BUFFER
1949 : - IMPLIED UNIT DATA BLOCK
1950 : - CSNRDY - TIMEOUT CONSTANT
1951 : OUTPUTS - R0 IS INCREMENTED.
1952 : ERROR - NOT-READY-TO-SEND TIME OUT
1953 :--
1954
1955 SNDBYT:: PUSH R1 ;ENTER R0-->BYTE
;MOV R1,-(SP)

1956 007102 010146 4$: MOV CSNRDY,R1 ;GET TIMEOUT CONSTANT FOR NOT READY ERROR
1957 007110 105775 1$: TSTB @XMSR(R5) ;READY TO SEND?
1958 007114 100412 BMI 2$ ;YES
1959 007116 010046 PUSH R0 ;NO, SAVE R0
;MOV R0,-(SP)

1960 007120 BREAK ;MONITOR BREAK
1961 007122 104422 POP R0 ;RESTORE
;MOV (SP)+,R0 TRAP CSBRK

1962
1963 007124 005301 DEC R1 ;ABORTED?
1964 007126 001370 BNE 1$ ;NO
1965 007130 012704 000056 MOV #TOSNDB,R4 ;YES,SET CODE FOR TIMEOUT ERROR
1966 007134 004737 012634 CALL LOG ;LOG IT
1967 007140 000402 BR 3$ ;QUIT
1968 007142 112075 000030 2$: MOVB (R0)+,@XMDB(R5) ;SEND IT
1969 007146 012601 3$: POP R1 ;RESTORE
;MOV (SP)+,R1

1970 007150 000207 RETURN ;DONE

```

```

1973 .SBTTL GETANS / GETS RESPONSES ROUND ROBIN USING 'XON'
1974
1975
1976 :++
1977 : GETANS - IF A UNIT IS RETRYING CLEAR HIS RECEIVE BUFFER (CLRBUF) AND GET
1978 : HIS RESPONSE (GTPKS1), ELSE, CLEAR ALL BUFFERS (CLRALL) AND
1979 : GET ALL RESPONSES (GTPKS8).
1980 : INPUTS: SYSTAT - SYSTEM STATUS WORD.
1981 : OUTPUTS: SERVST = -1 IF NO RETRIES.
1982 :--
1983
1984 007152 000240 GETANS:: NOP ;1 UNIT IF RETRY; ELSE ALL
1985 007154 032737 000006 003306 BIT #BIT1!BIT2,SYSTAT ;RETRY?
1986 007162 001010 BNE 1$ ;YES
1987 007164 012737 177777 010342 MOV #-1,SERVST ;PRESET NO UNITS SERVICED
1988 007172 004737 005710 CALL CLRALL ;CLEAR ALL INPUT BUFFERS
1989 007176 004737 007430 CALL GTPKS8 ;GET ALL REPLYs
1990 007202 000404 BR 2$ ;EXIT
1991 007204 004737 005750 1$: CALL CLRBUF ;RETRY-CLEAR 1 UNIT ONLY
1992 ;R5->UNIT BY NXTST
1993 007210 004737 007220 CALL GTPKS1 ;GET 1 REPLY
1994 007214 000207 2$: RETURN ;DONE
1995
1996 007216 000000 GETPTR: .WORD

```


1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055

.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
:          RSNDZ - END PACKET SIZE
:
:OUTPUTS: SYSTAT UPPER BYTE = FLAG BYTE RECEIVED
:--

```

```

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

```

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


```

2065 SBTTL GTPKS8 / GET RESPONSES (NO RETRIES)
2066
2067 :++
2068 : GTPKS8 - IF IN TEST 9 AND THE UNIT IS NOT MODIFIED, SKIP THE REST
2069 : OF THE ROUTINE. OTHERWISE:
2070 : SET ALL ABORTED UNITS SERVICED (SERVST: BIT POSITION). UNTIL
2071 : ALL UNITS SERVICED (SERVST=0). IF NO MORE PACKETS, SET UNIT
2072 : SERVICED, ELSE, GET A FLAG BYTE FROM UNIT, DECREMENTING THE
2073 : NUMBER OF PACKETS LEFT. CHECK TO SEE IF EXPECTED FLAG,
2074 : ADJUST COUNT IF NOT, GET REST OF PACKET. IF WAS DATA PACK,
2075 : AND NOT IN TEST 9, SEND 'XOFF' TO ENHANCE THROUGHPUT AND GO ON
2076 : TO NEXT UNIT (IF ANY). IF IN TEST 9, DO NOT SEND 'XOFFF'.
2077 :
2078 : INPUTS: (IMPLIED) UNITS DATA BLOCK POINTED TO BY R5. NONE PASSED.
2079 :          RSNDZ - END PACK SIZE
2080 :          RSDNSZ - DATA + END SIZE
2081 :
2082 : OUTPUTS: SYSTAT - UPPER BYTE=1ST BYTE RECEIVED, CURRENT UNIT
2083 : --
2084 007430 000240 GTPKS8:: NOP ;GET ALL UNITS RESPONSES XOFF IF DATA PAK (THROUGHPUT)
2085 007432 022737 000002 003342 CMP #2,TEST9 ;***** IS THIS TEST 9
2086 007440 001006 BNE 1$ ;***** NO
2087 007442 022765 000000 000210 CMP #0,MRSP(R5) ;***** IF SO, IS THIS UNIT MODIFIED
2088 007450 001002 BNE 1$ ;***** YES, CONTINUE NORMALLY
2089 007452 000137 010146 JMP ENDGP8 ;***** ELSE, SKIP ROUTINE
2090 007456 012737 003350 010344 1$: MOV #BLKTB,GTPT ;-->1ST
2091 007464 017705 000654 GTAGIN: MOV @GTPT,R5 ;GET DATA BLOCK
2092 007470 032715 100000 BIT #BIT15,@R5 ;ABORTED?
2093 007474 001404 BEQ 2$ ;NO
2094 007476 004737 010256 CALL SETSRV ;YES-SET' SERVICED' AND
2095 007502 000137 010104 JMP GTDOWN ;ON TO NEXT UNIT
2096 007506 105765 000033 2$: TSTB XSPKNT+1(R5) ;NO, ANY PACKETS LEFT?
2097 007512 001004 BNE 3$ ;YES
2098 007514 004737 010256 CALL SETSRV ;NO-HE'S DONE
2099 007520 000137 010104 JMP GTDOWN ;SO ON TO NEXT UNIT
2100 007524 105365 000033 3$: DECB XSPKNT+1(R5) ;NOW ITS ONE LESS PACKET
2101 007530 017537 000106 003314 MOV @XSPTR(R5),RCFLG ;GET EXPECTED FLAG
2102 007536 062765 000002 000106 ADD #2,XSPTR(R5) ;--> COUNT
2103 007544 017537 000106 003316 MOV @XSPTR(R5),RCBCNT ;AND EXPECTED COUNT
2104 007552 022737 000002 003342 CMP #2,TEST9 ;***** IF TEST 9
2105 007560 001404 BEQ 1$ ;***** DO NOT SEND XON
2106 007562 012700 007426 MOV #EXON,R0 ;--> XON
2107 ;***TIME CRITICAL
2108 007566 004737 007102 CALL SNDBYT ;***SEND IT
2109 007572 016500 000104 1$: MOV PKPTR(R5),R0 ;***-->WHERE 1ST BYTE GOES
2110 007576 004737 010346 CALL GTBYTE ;***GET IT
2111 007602 032715 100000 BIT #BIT15,@R5 ;ABORTED?
2112 007606 001404 BEQ 4$ ;NO-CONTINUE
2113 007610 105065 000033 CLRB XSPKNT+1(R5) ;YES-NO MORE PACKETS EXPECTED
2114 007614 000137 010104 JMP GTDOWN ;ON TO NEXT
2115 007620 005300 4$: DEC R0 ;-->BYTE JUST RECEIVED
2116 007622 111037 003307 MOVB @R0,SYSTAT+1 ;SAVE IT
2117 007626 121037 003314 CMPB @R0,RCFLG ;IS IT WHAT EXPECTED?
2118 007632 001436 BEQ GTOK ;YES
2119 007634 105065 000033 UNXPCT: CLRB XSPKNT+1(R5) ;NO, MUST BE LAST REPLY
2120 007640 121027 000002 CMPB @R0,#RSEND ;MAYBE AN END PAK?
2121 007644 001004 BNE 4$ ;NO

```

```

2122 007646 012737 000016 003316      MOV    #RSNDSZ,RCBCNT    ;YES, USE PROPER COUNT
2123 007654 000406                    BR      GTUM             ;AND GET IT
2124 007656 121027 000001      4$:    CMPB   @R0,#RSDATA    ;IS IT DATA?
2125 007662 001110                    BNE     GTDOWN        ;NO, ALL OVER, CHKANS WILL INIT UNIT
2126 007664 012737 000222 003316      MOV    #RSNSZ,RCBCNT    ;YES, USE COUNT OF DATA + END PAK SURE TO FOLLOW
2127 007672 005200                    GTUM:  INC     R0          ;WHERE TO STUFF THE REST
2128 007674 005337 003316      5$:    DEC     RCBCNT        ;ONE DOWN
2129 007700 001501                    BEQ     GTDOWN        ;NONE TO GO
2130 007702 004737 010346                    CALL   GTBYTE        ;MORE TO GO
2131 007706 032715 100000                    BIT    #BIT15,@R5      ;TIMEOUT?
2132 007712 001074                    BNE     GTDOWN        ;YES
2133 007714 005765 000074                    TST    DLV(R5)        ;BUT DLV ERROR?
2134 007720 001765                    BEQ     5$            ;NO
2135 007722 105065 000033                    CLRB   XSPKMM+1(R5)    ;YES-LAST TIME
2136 007726 000466                    BR      GTDOWN        ;ON TO NEXT
2137
2138 007730 005200      GTOK:  INC     R0          ;NEXT PLACE IN BUFFER
2139
2140      ;*****
2141 007732 022737 000002 003342      1$:    CMP    #2,TEST9      ;*** REV.- IF, NOT TEST 9
2142 007740 001022                    BNE     7$            ;*** REV.- THEN, NO MRSP HANDSHAKING REQUIRED
2143 007742 010046                    PUSH   R0            ;*** REV.- ELSE, TEST MRSP HANDSHAKE.
2144 007744 012737 000002 010254      MOV    #2,MRSPLY      ;*** REV.- DELAY FOR WAIT LOOP
2145 007752 005000      2$:    CLR     R0          ;*** REV.- THIS IS THE BEGINNING DELAY LOOP
2146 007754 005300      3$:    DEC     R0          ;*** REV.-
2147 007756 001376                    BNE     3$            ;*** REV.-
2148 007760 005337 010254                    DEC     MRSPLY      ;*** REV.-
2149 007764 001372                    BNE     2$            ;*** REV.- THIS IS THE END OF DELAY LOOP
2150
2151
2152 007766 105775 000022                    TSTB   @RCR(R5)        ;*** REV.- IF, DONE SET,
2153 007772 001066                    BNE     ERMOD        ;*** REV.- THEN, IT'S AN ERROR BECAUSE
2154
2155 007774 012700 010252      MOV    #MODRSP,R0      ;*** REV.- THERE WAS NO MRSP HANDSHAKE.
2156 010000 004737 007102      CALL   SNDBYT        ;*** REV.- ELSE, SEEMS TO BE OK, LETS
2157 010004 012600      POP     R0          ;*** REV.- SEND A 'CONTINUE' AND
2158
2159 010006 005337 003316      7$:    DEC     RCBCNT        ;*** REV.- SEE IF HANDSHAKE WORKS.
2160 010012 001413                    BEQ     4$            ;MORE BYTES?
2161 010014 004737 010346                    CALL   GTBYTE        ;NO-ALL DONE
2162 010020 032715 100000                    BIT    #BIT15,@R5      ;YES-GET IT
2163 010024 001027                    BNE     GTDOWN        ;TIMEOUT?
2164 010026 005765 000074                    TST    DLV(R5)        ;YES
2165 010032 001737                    BEQ     1$            ;ERROR?
2166 010034 105065 000033                    CLRB   XSPKMM+1(R5)    ;NO
2167 010040 000421                    BR      GTDOWN        ;LAST TIME
2168 010042 122775 000001 000104      4$:    CMPB   #RSDATA,@PKPTR(R5) ;EXIT
2169 010050 001015                    BNE     GTDOWN        ;WAS DATA?
2170 010052 010065 000104      MOV    R0,PKPTR(R5) ;NO, ALL DONE
2171
2172 010056 022737 000002 003342      ;***** ;START OF NEXT PACK NEXT TIME
2173 010064 001003      20$:   CMP    #2,TEST9      ;*** REV.- IF, TEST 9
2174

```

```

2174 010066 005765 000210          TST      MRSP(R5)      ;*** REV.- ANDIF, MRSP
2175 010072 001004                BNE      GTDOWN        ;*** REV.- THEN, NO HANDSHAKE
2176                                ;*****
2177 010074 012700 007427          20$:  MOV      #EXOFF,R0      ;XOFF AND SEND TO
2178 010100 004737 007102          CALL     SNDBYT          ;ENHANCE THROUGHPUT
2179 010104 062765 000002 000106 GTDOWN: ADD     #2,XSPTR(R5) ;NEXT XSFLG FOR NEXT TRY
2180 010112 023727 010344 003366      CMP      GTPTR,#LSTDEV ;DONE ONE CYCLE ALL UNITS?
2181 010120 103005                BHIS     1$              ;YES
2182 010122 062737 000002 010344      ADD      #2,GTPTR      ;NEXT UNIT
2183 010130 000137 007464          JMP      GTAGIN          ;CONTINUE RECEIVE
2184 010134 105737 010342          1$:  TSTB     SERVST        ;DONE SERVICING ALL PAKS
2185                                ;FROM ALL UNITS?
2186 010140 001402                BEQ      ENDGP8          ;YES
2187 010142 000137 007430          JMP      GTPKS8          ;NO, KEEP TRYING
2188 010146 000207          ENDGP8: RETURN ;RETURN
2189
2190 010150 000240          ERRMOD: NOP                     ;*** REV.- MRSP ERROR
2191 010152                PRINTF    #MESMRS,UNITNO
2192 010152 013746 027366                MOV      UNITNO,-(SP)
2193 010156 012746 010200                MOV      #MESMRS,-(SP)
2194 010162 012746 000002                MOV      #2,-(SP)
2195 010166 010600                MOV      SP,R0
2196 010170 104417                TRAP     C$PNTF
2197 010172 062706 000006                ADD      #6,SP
2198 010176 000207          RETURN
2199
2194 010200 045 116 045 MESMRS: .ASCIZ  '!%N%S9%S2%01%S9%S9%AERROR IN MRSP PROTOCOL'
2195                                .EVEN
2196 010252 020 MODRSP: .BYTE  RSCONT
2197                                .EVEN
2198 010254 000000 MRSDLY: .WORD

```

```
2201 .SBTTL SETSRV / SET UNIT SERVICED
2202
2203 :++
2204 : SETSRV - RESET THE BIT IN 'SERVST' CORRESPONDING TO THE UNIT NUMBER.
2205 : INPUTS - SERVST - 'SERVICED' WORD
2206 : - @R5 = UNIT # (BITS 0, 1, 2)
2207 : OUTPUTS - SERVST MODIFIED
2208 :--
2209
2210 010256 010546 SETSRV: PUSH R5 ;SET UNIT SERVICED
MOV R5,-(SP)

2211 010260 010046 PUSH R0 MOV R0,-(SP)

2212 010262 011505 MOV @R5,R5 ;GET STAT WD
2213 010264 042705 177770 BIC #177770,R5 ;MASK UNIT #
2214 010270 012700 010322 MOV #SRVTBL,R0 ;->TOP OF BIT TABLE
2215 010274 005705 1$: TST R5 ;RIGHT ONE?
2216 010276 001404 BEQ 2$ ;YES
2217 010300 062700 000002 ADD #2,R0 ;NO, ->NEXT
2218 010304 005305 DEC R5 ;1 LESS
2219 010306 000772 BR 1$ ;CONTINUE
2220 010310 041037 010342 2$: BIC @R0,SERVST ;MOW IT DOWN
2221 010314 POP R0 MOV (SP)+,R0
010314 012600

2222 010316 POP R5 MOV (SP)+,R5
010316 012605

2223 010320 000207 RETURN ;RETURN
2224
2225 010322 000001 SRVTBL: .WORD BIT0 ;BIT POSITION LOOKUP TABLE
2226 010324 000002 .WORD BIT1
2227 010326 000004 .WORD BIT2
2228 010330 000010 .WORD BIT3
2229 010332 000020 .WORD BIT4
2230 010334 000040 .WORD BIT5
2231 010336 000100 .WORD BIT6
2232 010340 000200 .WORD BIT7
2233
2234 010342 000000 SERVST: .WORD
2235 010344 000000 GTPTR: .WORD
```

```

2238 .SBTTL GTBYTE / GET A BYTE FROM UNIT
2239
2240 :++
2241 :GTBYTE - TEST INTERFACE FOR 'READY-TO-RECEIVE' AND INPUT A BYTE, IF
2242 :SO. IF NOT, THE FOLLOWING OCCURS: SEND 'XOFF' TO UNIT IN
2243 :PREPARATION FOR ^C CHECK ('BREAK' TO SUPERVISOR). WAIT
2244 :TO SEE IF A CHARACTER SLOPS OVER DUE TO UART LATENCY. IF
2245 :ONE DOES THEN MIGHT AS WELL GET IT AND SEND 'XON' TO GET
2246 :THE REST OF THE MESSAGE, OTHERWISE, 'BREAK'. THEN SEND
2247 :'XON', AND TEST FOR LONG TIMEOUT (A 30 SECOND REWIND). IF SO,
2248 :LOG ERROR, OTHERWISE REPEAT THE ABOVE UNTIL READY OR TIME OUT.
2249 :REMEMBER TO PRESERVE R0 SINCE THE 'BREAK' TRAP CLOBBERS IT.
2250
2251 :INPUTS - R0 POINTS TO INPUT BUFFER
2252 :        - IMPLIED UNITS DATA BLOCK
2253 :        - CSRCVB TIME OUT MULTIPLIER
2254
2255 :OUTPUTS - R0 IS INCREMENTED
2256 :        - DLV (R5) NON-ZERO ON INTERFACE ERROR.
2257
2258 :ERROR - TIME OUT ON RECEIVE
2259 :--
2260
2261 010346 005037 010572 GTBYTE:: CLR GBTMP ;TIMEOUT REGISTER
2262 010352 013704 003346      MOV CSRCVB,R4 ;TIMEOUT ERROR CONSTANT (MULTIPLIER)
2263 010356 105775 000022 1$: TSTB @RCSR(R5) ;READY?
2264 010362 100013          BPL 3$ ;NO
2265 010364 017565 000024 000074 MOV @RCDB(R5),DLV(R5) ;GET ERROR + BYTE
2266 010372 116520 000074      MOVB DLV(R5),(R0)+ ;COPY BYTE TO BUFFER
2267 010376 005765 000074      TST DLV(R5) ;ERROR?
2268 010402 100472          BMI 4$ ;YES-EXIT
2269 010404 005065 000074      CLR DLV(R5) ;NO-RESET
2270 010410 000467          BR 4$ ;AND EXIT
2271 010412 005337 010572 3$: DEC GBTMP ;DEC T.O. CONSTANT
2272 010416 001357          BNE 1$ ;STILL VALID
2273
2274 ;CODE TO SEE ^C DURING LONG SEEK OR REWIND
2275      MOV R0,GBTMP2 ;HERE GBTMP=0
2276 010420 010037 010574      MOV #EXOFF,R0 ;R0 MUST BE PRESERVED!
2277 010424 012700 007427      CALL SNDBYT ;QUIET THE DEVICE
2278 010430 004737 007102      TSTB @RCSR(R5) ;BY SENDING XOFF
2279 010434 105775 000022 6$: BMI 5$ ;CHARACTER SLOP OVER?
2280 010440 100415          DEC GBTMP ;YES
2281 010442 005337 010572      TSTB GBTMP ;NO-WAIT A WHILE
2282 010446 105737 010572      BNE 6$ ;DONE WAITING?
2283 010452 001370          BREAK ;NO
2284 010454          ;YES-NO SLOP OVER
2285 010456 104422          TRAP CSBRK
2286 010462 012700 007426      MOV #EXON,R0 ;START DEVICE TALKING
2287 010466 004737 007102      CALL SNDBYT ;AGAIN
2288 010472 013700 010574      MOV GBTMP2,R0 ;RESTORE R0
2289 010474 000426          BR 7$ ;END KLUGE
2290 010500 013700 010574 5$: MOV GBTMP2,R0 ;RESTORE R0
2291 010506 017565 000024 000074 MOV @RCDB(R5),DLV(R5) ;GET ERROR + BYTE
2292 010512 116520 000074      MOVB DLV(R5),(R0)+ ;COPY BYTE TO BUFFER
2293 010516 005765 000074      TST DLV(R5) ;ERROR?
2294          BMI 17$ ;YES-EXIT

```

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


```

2311 .SBTTL CHKANS / CHECK DEVICE(S) RESPONSE
2312
2313 :++
2314 :CHKANS - AS IN 'GETANS', IF RETRYING DO ONLY 1 UNIT ELSE DO ALL NON-
2315 :ABORTED UNITS. NOTE, IF IN TEST 9 AND THE UNIT IS NOT
2316 :MODIFIED DO NOT CHECK UNIT.
2317 :INPUTS: IMPLIED SYSTAT BIT1 (RETRYING)
2318 :BLKTB1 - TOP OF DATA BLOCK ALLOCATION TABLE
2319 :LSTDEV - ADDR. OF LAST UNIT'S DATA BLOCK
2320
2321 :OUTPUTS: NONE PASSED.
2322 :--
2323
2324 010576 000240 CHKANS:: NOP ;IF RETRY THEN CHECK ONE
2325 ;ELSE CHECK ALL
2326 010600 032737 000006 003306 BIT #BIT1!BIT2,SYSTAT ;RETRYING?
2327 010606 001403 BEQ CHK8 ;NO DO NORMAL
2328 010610 004737 010706 CALL CHKPXS ;YES DO SINGLE UNIT
2329 ;RS -> UNIT
2330 010614 000432 BR CHKANR ;ALL DONE
2331
2332 010616 012737 003350 010704 CHK8: MOV #BLKTB1,CHKPTR ;YOU KNOW... TOP OF TABLE
2333 010624 017705 000054 2$: MOV @CHKPTR,RS ;GET UNIT'S BLOCK ADDRESS
2334 010630 032715 100000 BIT #BIT15,@RS ;ABORTED?
2335 010634 001012 BNE 3$ ;YES
2336 010636 022737 000002 003342 CMP #2,TEST9 ;***** IS THIS TEST 9
2337 010644 001004 BNE 1$ ;***** NO-CONTINUE NORMALLY
2338 010646 022765 000000 000210 CMP #0,MRSP(RS) ;***** IF SO, IS THIS UNIT MODIFIED
2339 010654 001402 BEQ 3$ ;***** NO SKIP NEXT INSTR
2340 010656 004737 010706 1$: CALL CHKPXS ;NO, DO THIS GUY
2341 010662 023727 010704 003366 3$: CMP CHKPTR,#LSTDEV ;ALL DONE?
2342 010670 103004 BHS CHKANR ;YES
2343 010672 062737 000002 010704 ADD #2,CHKPTR ;NO,--> NEXT DEVICE
2344 010700 000751 BR 2$ ;DO IT
2345
2346 010702 000207 CHKANR: RETURN
2347
2348 010704 000000 CHKPTR: .WORD

```

```

2351 .SBTTL CHKPKS / DECIPHERS RESPONSE OF UNIT POINTED TO BY R5 /
2352
2353 :++
2354 :CHKPKS - FOR UNIT R5 AND FOR ALL PACKETS, CHECK TO SEE IF PACKET IS DATA OR
2355 :END PACK, CHECK CHECKSUMS, COMPARE DATA IF DATA PACK, CHECK
2356 :SUCCESS CODE IF END. IF UNKNOWN PACKET TYPE, CHECK FOR INTERFACE
2357 :ERROR. IF "CONTINUE" FALL THROUGH. IF "INIT" SET "SEND
2358 :BREAK" FLAG. CALL "LOG" WITH R4=ERROR NUMBER IF ERROR.
2359 :THIS ROUTINE IS ALSO USED TO DETERMINE THE PROTOCOL OF A UNIT. IN
2360 :THE FIRST PART OF TEST 9 A GET CHARACTERISTICS COMMAND PACKET WAS
2361 :SENT TO THE TU58. IF THE RESPONSE WAS A DATA PACKET, WHICH IS
2362 :EXPECTED, THEN THE UNIT IS NOT MODIFIED, AND THE MRSP FLAG IS
2363 :CLEARED. IF THE RESPONSE IS AN END PACKET, WHICH WOULD BE
2364 :HANDLED BY THIS ROUTINE AS AN UNKNOWN, THEN THE UNIT IS MODIFIED,
2365 :AND THE MRSP FLAG IS SET.
2366 :INPUTS: (IMPLIED) UNITS DATA BLOCK
2367
2368 :OUTPUTS: ERRORS - DLV ERROR
2369 :              - UNKNOWN FLAG BYTE ERROR
2370 :              - CHECKSUM ERROR
2371 :              - DATA COMPARE ERROR
2372 :R4 = ERROR NUMBER
2373 :SYSTAT UPPER BYTE = 1ST BYTE OF RESPONSE
2374 :--
2375
2376 010706 000240
2377 010710 042715 000010
2378 010714 016500 000102
2379 010720 116502 000032
2380 010724 012703 000034
2381 010730 060503
2382 010732 010301
2383 010734 062701 000002
2384 010740 010065 000104
2385 010744 111037 003307
2386 010750 011137 003316
2387 010754 011337 003314
2388 010760 121013
2389 010762 001057
2390 010764 121027 000020
2391 010770 001534
2392
2393 010772 013704 003316
2394 010776 005744
2395 011000 004737 013750
2396 011004 103005
2397 011006 012704 000022
2398 011012 004737 012634
2399 011016 000521
2400 011020 122710 000002
2401 011024 001005
2402 011026 004737 011302
2403 011032 012702 000001
2404 011036 000511
2405 011040 122710 000001
2406 011044 001012
2407 011046 022737 000001 003342

```

```

CHKPKS:: NOP
          BIC      #BIT3,R5      ;CHECK WHAT WAS RECIEVED
          MOV      RCVBUF(R5),R0 ;CLEAR 'BAD FLAG' RETRY BIT
          MOVB     XSPKNT(R5),R2 ;GET BUFFER ADDR.
          MOV      #XSFLG,R3     ;AND # OF PACKETS EXPECTED
          ADD      R5,R3         ;THE OFFSET VALUE
          MOV      R3,R1         ;R3-->THIS UNIT XSFLG AGAIN
          ADD      #2,R1         ;COPY TO R1
          MOV      R0,PKPTR(R5) ;R1-->XSBCNT FOR 1ST PACKET
          MOVB     @R0,SYSTAT+1 ;POINT TO PACKET
          MOV      @R1,RCBCNT    ;SAVE RCV'D BYTE
          MOV      @R3,RCFLG     ;GET COUNT
          CMPB     @R0,@R3       ;AND FLAG
          BNE      5$            ;1ST BYTE=EXPECTED?
          CMPB     @R0,#RSCONT   ;UH OH...
          BEQ      7$            ;OK, IS IT 1 BYTE?
          MOV      RCBCNT,R4     ;YES...ONTO NEXT PACK
          TST      -(R4)         ;NO, SO > 1 BYTE (NEVER EXPECT INIT!)
          CALL     CKCKSM        ;EXPECTED, SO COUNT MUST BE RIGHT
          BCC      2$            ;ADJUST FROM RECEIVE COUNT TO COUNT FOR CHECKSUM
          MOV      #BDCHK,R4     ;CHECK CHECKSUM
          CALL     LOG           ;NO CARRY...NO INCORRECT
          BR       7$           ;ERROR
          CMPB     #RSEND,(R0)  ;LOG IT
          BNE      3$            ;ON TO NEXT PACK
          CALL     CHKEND        ;END PAK?
          BR       7$           ;NO
          CMPB     #RSDATA,@R0  ;YES-CHECK
          BNE      4$            ;LAST PACKET
          CMP      #1,TEST9      ;AND FALL THROUGH
          ;***** IS THIS TEST 9

```



```

2408 011054 001003          BNE      11$      :***** NO-CONTINUE NORMALLY
2409 011056 005065 000210   CLR      MRSP(R5) :***** CLR MRSP FLAG
2410 011062 000402          BR       12$      :***** SKIP INSTR
2411 011064 004737 014550   11$: CALL    COMPAR :YES-CHECK DATA
2412 011070 000474          12$: BR       7$      :ALL DONE?
2413 011072 052715 020010   4$: BIS     #BIT3!BIT13,R5 :SET 'BAD FLAG' RETRY FLAGS
2414 011076 012704 000010   MOV     #OTL,R4 :OUT TO LUNCH
2415 011102 005765 000074   TST     DLV(R5) :AH,BUT DLV ERROR?
2416 011106 001402          BEQ     20$      :NO
2417 011110 012704 000012   MOV     #OVRN,R4 :YES-USE CORRECT ERROR #
2418 011114 004737 012634   20$: CALL    LOG :TALLY
2419 011120 000467          BR       8$      :DONE
2420
2421          :HERE CHECKS UNEXPECTED RESPONSE
2422
2423 011122 122710 000004   5$: CMPB    #RSINIT,R0 :INIT?
2424 011126 001007          BNE     6$      :NO
2425 011130 052715 020010   BIS     #BIT3!BIT13,R5 :YES-SET RETRY FLAGS
2426 011134 012704 000006   MOV     #RCINIT,R4 : WE GOT AN INIT
2427 011140 004737 012634   CALL    LOG :TALLY IT
2428 011144 000455          BR       8$      :DONE
2429 011146 122710 000001   6$: CMPB    #RSDATA,R0 :DATA PAK?
2430 011152 001013          BNE     9$      :NO
2431 011154 012704 000204   MOV     #RSDASZ,R4 :YES, USE DATA SIZE
2432 011160 005744          TST     -(R4) :ADJUST FOR CHKSUM
2433 011162 004737 013750   CALL    CKCKSM :AND CHECK
2434 011166 103430          BCS     10$     :GOOF
2435 011170 004737 014550   CALL    COMPAR :OK, HOW'S THE DATA?
2436          :EXPECTED END, GOT
2437          :DATA + END.
2438 011174 062700 000204   ADD     #RSDASZ,R0 :POINT TO END PACK
2439 011200 000657          BR       1$      :CHECK IT, USE SAME XSFLG
2440
2441 011202 122710 000002   9$: CMPB    #RSEND,(R0) :END?
2442 011206 001331          BNE     4$      :NO-OUT TO LUNCH
2443 011210 012704 000016   MOV     #RSSNSZ,R4 :YES, TOTAL SIZE MINUS
2444 011214 005744          TST     -(R4) :TWO (THE CHKSUM)
2445 011216 004737 013750   CALL    CKCKSM :CHECK IT
2446 011222 103412          BCS     10$     :OOPS
2447 011224 022737 000001 003342   CMP     #1,TEST9 :***** IS THIS TEST 9
2448 011232 001003          BNE     13$     :***** NO-CONTINUE NORMALLY
2449 011234 012765 000001 000210   MOV     #1,MRSP(R5) :***** IF SO, SET THE MRSP FLAG
2450 011242 004737 011302   13$: CALL    CHKEND :OK,NOW TEST SUC. CODE
2451
2452 011246 000414          BR       8$      :ALL DONE
2453
2454 011250 012704 000022   10$: MOV     #BDCHK,R4 :CHECKSUM ERROR
2455 011254 004737 012634   CALL    LOG
2456 011260 000407          BR       8$      :EXIT
2457
2458 011262 005302          7$: DEC     R2 :ANY PACKETS LEFT TO CHECK?
2459 011264 001405          BEQ     8$      :NO, ALL DONE
2460 011266 063700 003316   ADD     RBCNT,R0 :YES, POINT TO NEXT PACKET
2461 011272 022121          CMP     (R1)+,(R1)+ :POINT TO NEXT EXPECTED COUNT
2462 011274 022323          CMP     (R3)+,(R3)+ :AND EXPECTED FLAG
2463 011276 000620          BR       1$      :TRY ANOTHER,THEY'RE SMALL
2464 011300 000207          8$: RETURN :RETURN

```

```

2467 .SBTTL CHKEND / CHECK SUCCESS AND DETERMINE RETRY STATUS /
2468
2469 :++
2470 :CHKEND - IF RETRYING;DETERMINE IF DATA ERROR OR BAD.FLAG BYTE ERROR RETRY.
2471 :
2472 :   IF RETRYING BAD FLAG:RESET RETRY FLAG(SINCE OPERATION IS COMPLETE),
2473 :   AND CHECK SUCCESS CODE.
2474 :   IF RETRYING DATA ERROR ; CHECK SUCCESS CODE AND IF 0, PRINT RECOVERED, LOG
2475 :   SOFT ERROR, END RETRY STATUS. IF NOT 0 AND WAS STILL 'DATA
2476 :   CHECK' ERROR - DETERMINE WHETHER TO CONTINUE ANOTHER RETRY OR
2477 :   LOG 'UNRECOVERABLE' ERROR.
2478 :
2479 :   IF NOT RETRYING DATA ERROR; CHECK IF 'DATA CHECK' ERROR SUCCESS CODE,
2480 :   AND IF SO,START RETRY, ELSE EXIT.
2481 :
2482 :   INPUTS: IMPLIED UNITS DATA BLOCK
2483 :   OUTPUTS: RETRY (SYSTAT BIT 1AND 2), (BIT10 @R5) RESET IF RETRYING.
2484 :             - DATA COMARE ERROR (BIT6 @R5) CLEARED.
2485 :             - REDUCED/NORMAL GAIN (BIT7 @R5) ADJUSTED
2486 :--
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
011302 000240
011304 010046
011306 010446
011310 032737 000006 003306 1$: BIT #BIT1!BIT2,SYSTAT ;RETRYING?
011316 001406 BEQ NOREE ;NO-CHECK NORMALLY
011320 032737 000004 003306 BIT #BIT2,SYSTAT ;IS IT BAD FLAG TYPE?
011326 001454 BEQ CHKREE ;NO(DATA TYPE)
011330 042715 020000 BIC #BIT13,@R5 ;YES, SO IF END PACK THEN RETRY'S COMPLETE
011334 004737 012320 NOREE: CALL CHKSUC ;CHECK SUCCESS CODE
011340 032715 100000 BIT #BIT15,@R5 ;ABORTED?
011344 001402 BEQ 3$ ;NO,CONTINUE
011346 000137 012024 JMP CHKRET ;YES,EXIT
011352 105765 000077 3$: TSTB SUCCS+1(R5) ;NO; HOW'D WE DO?
011356 001013 BNE CHKERR ;NOT SO GOOD.
011360 032715 000100 BIT #BIT6,@R5 ;OK, HOST FIND DATA PAK ERROR?
011364 001002 BNE 2$ ;YES
011366 000137 012024 JMP CHKRET ;NO
011372 012704 000014 2$: MOV #BDCOM,R4 ;YES; JUST BAD DATA-NO DATACHK ERR
011376 004737 012634 CALL LOG ;BAD DATA IN PACKET
011402 000137 012024 JMP CHKRET ;QUIT
011406 032715 001000 CHKERR: BIT #BIT9,@R5 ;BAD SUCCESS; TU DATA CHK ERROR?
011412 001002 BNE 1$ ;YES
011414 000137 012024 JMP CHKRET ;NO. ALL DONE.
011420 052715 002000 1$: BIS #BIT10,@R5 ;YES-START RETRY
011424 012765 000001 000002 MOV #1,RETRY(R5) ;CALL IT 1ST
011432 PRINTX #RTRYN,RETRY(R5) ;** PRINT **
011432 016546 000002 MOV RETRY(R5),-(SP)
011436 012746 012204 MOV #RTRYN,-(SP)
011442 012746 000002 MOV #2,-(SP)
011446 010600 MOV SP,R0
011450 104415 TRAP C$PNTX

```

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

```

011716 104415
011720 062706 000006
2544 011724 032715 000200
2545 011730 001403
2546 011732 042715 000200
2547 011736 000432
2548 011740 052715 000200
2549 011744 000427
2550 011746 000240
2551 011750
011750 012746 012224
011754 012746 000001
011760 010600
011762 104415
011764 062706 000004
2552 011770 032715 000400
2553 011774 001003
2554 011776 012704 000016
2555 012002 000402
2556 012004 012704 000020
2557 012010 004737 012634
2558 012014 005065 000002
2559 012020 042715 002200
2560
2561 012024 042737 000006 003306
2562 012032 042715 000100
2563 012036
012036 012604
2564 012040
012040 012600
2565 012042 000207
2566
2567
2568 012044 045 101 122 RECOV: .ASCIZ /%ARECOVERED FROM DATA CHECK ERROR RETRY # %D1%N/
2569 .EVEN
2570 012124 045 101 040 THRSLO: .ASCIZ /%A NORMAL THRESHOLD%N/
2571 .EVEN
2572 012152 045 101 040 THRSHI: .ASCIZ /%A MODIFIED THRESHOLD %N/
2573 .EVEN
2574 012204 045 101 122 RTRYN: .ASCIZ /%ARETRY # %D1%N/
2575 .EVEN
2576 012224 045 101 125 UNREC: .ASCIZ /%AUNRECOVERABLE%N/
2577 .EVEN
2578 012246 045 101 117 RETERR: .ASCIZ /%AOTHER ERROR DURING RETRY : EXIT RETRY%N/
2579 .EVEN

1$:
HRD1:
4$:
5$:

BIT #BIT7,@R5 ;WAS NORMAL THRESHOLD?
BEQ 1$ ;YES-REDUCE GAIN
BIC #BIT7,@R5 ;NO-NORM
BR CHKRET
BIS #BIT7,@R5 ;REDUCED
BR CHKRET ;DONE
NOP ;HERE IS HARD ERROR!
PRINTX #UNREC

TRAP ADD C$PNTX #6,SP

MOV #UNREC,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTX #4,SP
ADD

BIT #BIT8,@R5 ;RD OR WR?
BNE 4$ ;WRITE
MOV #HRDRD,R4 ;READ
BR 5$ ;LOG IT
MOV #HRDWR,R4 ;WRITE
CALL LOG ;LOG IT
CLR RETRY(R5) ;BACK TO NORMAL
BIC #BIT10!BIT7,@R5 ;NO RETRY, NOT REDUCED

CHKRET: BIC #BIT1!BIT2,SYSTAT ;NO SYSTEM RETRY NEXT PASS
BIC #BIT6,@R5 ;NO MORE HOST DATA CHECK ERROR
POP R4
MOV (SP)+,R4
POP R0
MOV (SP)+,R0
RETURN
```

```
2582 .SBTTL CHKSUC / INTERPRET SUCCESS CODE /
2583
2584 :++
2585 : CHKSUC - COPY SUCCESS CODE (BYTE) TO SUCCS+1(R5). INTERPRET SUCCESS
2586 : AND IF NOT 0, LOG APPROPRIATE ERROR.
2587 : INPUTS: R0 POINTS TO END PACKET.
2588 : @R5 - UNIT STATUS WORD
2589 : CMDSNT(R5) - COMMAND BYTE
2590 :
2591 : OUTPUTS: R4 IS ERROR NUMBER IF ERROR.
2592 : SUCCS(R5) UPDATED.
2593 : BIT9 @R5 SET ON DATA CHECK SUCCESS CODE
2594 :--
2595
2596 012320 000240 CHKSUC:: NOP
2597 012322 016065 000002 000076 MOV 2(R0),SUCCS(R5) ;R0-->END PACKET
2598 012330 122760 000000 000003 CMPB #ESOK,3(R0) ;GET SUCCESS BYTE
2599 012336 001535 BEQ 12$ ;COMPLETE SUCCESS-EXIT
2600
2601 012340 122760 000001 000003 CMPB #ESTRY,3(R0) ;OK BUT RETRIES?
2602 012346 001012 BNE 20$ ;NO
2603 012350 126527 000100 000002 CMPB CMDSNT(R5),#RSSRD ;A READ?
2604 012356 001001 BNE 22$ ;NO
2605
2606 012360 000520 BR 10$ ;NO RETRIES IN MAINTENANCE!
2607 012362 126527 000100 000003 22$: CMPB CMDSNT(R5),#RSSWR ;A WRITE?
2608 012370 001001 BNE 20$ ;NO
2609 012372 000513 BR 10$ ;LOG IT
2610 012374 122760 177737 000003 20$: CMPB #ESNOMO,3(R0) ;NO MOTOR?
2611 012402 001003 BNE 1$ ;NO
2612 012404 012704 000030 MOV #NOMOT,R4 ;YES-
2613 012410 000506 BR 11$ ;LOG
2614
2615 012412 122760 177757 000003 1$: CMPB #ESCKS,3(R0) ;'DATA CHECK' ERROR?
2616 012420 001003 BNE 2$ ;NO
2617 012422 052715 001000 BIS #BIT9,@R5 ;SET DATA-CHK-ERROR FLAG
2618 012426 000501 BR 12$ ;DONT LOG
2619
2620 012430 126527 000100 000007 2$: CMPB CMDSNT(R5),#RSSSLF ;SELF TEST?
2621 012436 001006 BNE 3$ ;NOPE
2622 012440 105760 000003 TSTB 3(R0) ;YES, NEG. IF ERROR
2623 012444 100072 BPL 12$ ;OK
2624
2625 012446 012704 000044 MOV #SLFER,R4 ;YES-ERROR
2626 012452 000465 BR 11$ ;LOG IT
2627
2628 012454 122760 177740 000003 3$: CMPB #ESSK,3(R0) ;SEEK ERROR?
2629 012462 001005 BNE 4$ ;NO
2630 012464 012704 000024 MOV #SKERR,R4 ;YES-
2631 012470 052705 040000 BIS #BIT14,R5 ;SET 'DOBRK' FLAG
2632 012474 000454 BR 11$ ;LOG
2633
2634 012476 122760 177767 000003 4$: CMPB #ESNCRT,3(R0) ;NO CART?
2635 012504 001003 BNE 5$ ;NO
2636 012506 012704 000054 MOV #NCART,R4 ;YES-
2637 012512 000445 BR 11$ ;LOG
2638
```

2639	012514	122760	177720	000003	5\$:	CMPB	#ESCMD,3(R0)	;NO UNDERSTAND HOST?
2640	012522	001003				BNE	6\$	;NO
2641	012524	012704	000040			MOV	#CMNDR,R4	;YES-
2642	012530	000436				BR	11\$	;LOG
2643								
2644	012532	122760	177770	000003	6\$:	CMPB	#ESNONX,3(R0)	;NON EXISTENT UNIT?
2645	012540	001003				BNE	7\$	;NO
2646	012542	012704	000036			MOV	#NUNIT,R4	;YES-
2647	012546	000427				BR	11\$	;LOG
2648								
2649	012550	122760	177765	000003	7\$:	CMPB	#ESWLOC,3(R0)	;WRITE LOCKED?
2650	012556	001003				BNE	8\$	;NO
2651	012560	012704	000026			MOV	#WRLOCK,R4	;YES-
2652	012564	000420				BR	11\$	;LOG
2653								
2654	012566	122760	177776	000003	8\$:	CMPB	#ESPART,3(R0)	;PARTIAL OP?
2655	012574	001003				BNE	9\$	;NO
2656	012576	012704	000034			MOV	#PARTL,R4	;YES-
2657	012602	000411				BR	11\$	;LOG
2658								
2659	012604	122760	177711	000003	9\$:	CMPB	#ESREC,3(R0)	;WRONG RECORD?
2660	012612	001003				BNE	10\$	;NO
2661	012614	012704	000042			MOV	#RECERR,R4	;YES-
2662	012620	000402				BR	11\$	;LOG
2663								
2664	012622	012704	000046		10\$:	MOV	#SUCCTL,R4	;UNDEFINED
2665	012626	004737	012634		11\$:	CALL	LOG	;LOG ERROR
2666	012632	000207			12\$:	RETURN		;RETURN


```

2669 .SBTTL LOG / TO LOG ERROR IN CORRECT PLACE
2670
2671 :++
2672 : LOG - DETERMINE IF ERROR IS FATAL, NON-FATAL OR FATAL AFTER N TRIES
2673 : BY INDEX (ERROR #) INTO DEVICE DATA BLOCK. ADD THE DRIVE # TO
2674 : INDICATE UPPER OR LOWER BYTE AND INCREMENT THAT ERROR UNLESS
2675 : THAT BYTE WOULD OVERFLOW. DETERMINE IF EVL FLAG SET, AND IF SO,
2676 : CHECK THRESHOLD (EVLTHR) AND PRINT APPROPRIATE ERROR MESSAGE
2677 : DESCRIPTION. ABORT THE UNIT IF INDICATED THROUGH DODROP CODE.
2678 : INPUTS: R4 = ERROR CODE
2679 : OUTPUTS: ABNDX(R5) = ERROR CODE.
2680 :          DLV(R5) = 0
2681 :          L$LUN = UNIT NUMBER
2682 :--
2683
2684 012634 010046 LOG:: PUSH R0 MOV R0,-(SP)
012634
2685 012636 010146 PUSH R1 MOV R1,-(SP)
012636
2686 012640 010346 PUSH R3 MOV R3,-(SP)
012640
2687 012642 010446 PUSH R4 MOV R4,-(SP)
012642

2688
2689 012644 011537 002074 MOV @R5,L$LUN ;GET UNIT NUMBER
2690 012650 042737 177770 002074 BIC #177770,L$LUN ;MASK IT OFF
2691 012656 010465 000004 MOV R4,ABNDX(R5) ;SAVE INDEX IN CASE OF ABORT MESSAGE
2692 012662 012703 000120 MOV #LGOFST,R3 ;OFFSET TO LOW ORDER BYTE (DRIVE0)
2693 012666 060403 ADD R4,R3 ;FORM INDEX OF PARAM. TO UPDATE
2694 012670 060503 ADD R5,R3 ;FORM ABSOLUTE ADDR. THIS UNIT
2695 012672 004737 013640 CALL WHCHDR ;SEE WHICH DRIVE T'WAS
2696 012676 103001 BCC 2$ ;WAS DRIVE 0
2697 012700 005203 INC R3 ;DRIVE 1: POINT TO UPPER BYTE
2698 012702 122713 000377 2$: CMPB #255.,@R3 ;POTENTIAL OVERFLOW POSSIBLE?
2699 012706 001005 BNE LOGOK ;NO
2700 012710 LOGO: ERRDF 0.,OVRFLO,ERRDES ;YES

012710 104455 TRAP CSERDF
012712 000000 .WORD 0
012714 013534 .WORD OVRFLO
012716 013170 .WORD ERRDES

2701 012720 000512
2702 012722 105213 LOGOK: BR ABO ;ABORT UNIT
2703 012724 111304 INCB @R3 ;INCREMENT THE ERROR
2704 012726 016503 000004 MOV @R3,R4 ;TEMP'LY SAVE IT
2705 012732 012701 002226 MOV ABNDX(R5),R3 ;GET INDEX AGAIN
2706 012736 066501 000004 *JV #RSNTAB,R1 ;FORM ADRS OF MSG
2707 012742 042701 000001 ADD ABNDX(R5),R1 ;LIKE THIS
2708 012746 032737 000004 016752 BIC #BIT0,R1 ;INSURE WC'D BOUNDARY
2709 012754 001414 BIT #EVL,FLGLOC ;EVL SELECTED?
BEQ LOGOK2 ;NO-CONT

```

```
2710 012756 123704 002222          CMPB   EVLTHR,R4      ;YES,OVER THRESHOLD?
2711 012762 101011          BHI     LOGOK2        ;NO
2712 012764 010337 012776          MOV     R3,DFTL1+2    ;YES,LOAD ERROR #
2713 012770 011137 013000          MOV     @R1,DFTL1+4    ;AND MESSAGE ADDR
2714 012774          ERRDF   0,DFTL1,ERRDES ;ERROR
                                DFTL1:
                                012774 104455
                                012776 000000
                                013000 012774
                                013002 013170
2715 013004 000460          BR       ABO             ;DROP IT
2716 013006 120327 000014          LOGOK2: CMPB   R3,#BDCOM    ;'NEVER FATAL' TYPE?
2717 013012 103011          BHS     NTSFT          ;NO
2718 013014 010337 013026          MOV     R3,LOG1+2    ;YES, ERROR CODE
2719 013020 011137 013030          MOV     @R1,LOG1+4    ;DESCRIPTION
2720 013024          ERRSFT  0.,LOG1,ERRDES
                                LOG1:
                                013024 104457
                                013026 000000
                                013030 013024
                                013032 013170
2721 013034 000450          BR       LOGO             ;EXIT
2722
2723 013036 120327 000026          NTSFT: CMPB   R3,#WRLOCK    ;ONE TRY?
2724 013042 103411          BLO     MABEE          ;NO, MAYBE A MULTIPLE
2725 013044 010337 013056          MOV     R3,LOG2+2.    ;YES
2726 013050 011137 013060          MOV     @R1,LOG2+4
2727 013054          ERRHRD  0,LOG2,ERRDES ;PRINT HARD MESSAGE
                                LOG2:
                                013054 104456
                                013056 000000
                                013060 013054
                                013062 013170
2728 013064 000430          BR       ABO             ;DROP UNIT
2729
2730 013066 042704 177400          MABEE: BIC     #177400,R4    ;NEGATE SIGN EXTEND
2731 013072 163704 003320          1$:  SUB     FTLNM,R4      ;SEE IF MULTIPLE OF
2732 013076 001413          BEG     HRD             ;FTLNM-YES!
2733 013100 103401          BLO     SFT             ;NO
2734 013102 000773          BR       1$             ;NOT THERE YET
2735
2736 013104 010337 013116          SFT:  MOV     R3,LOG3+2    ;ERROR CODE
2737 013110 011137 013120          MOV     @R1,LOG3+4    ;DESCRIPTION
2738 013114          ERRSFT  0,LOG3,ERRDES
                                LOG3:
                                013114 104457
                                013116 000000
                                013120 013114
                                013122 013170
2739 013124 000414          BR       LOGO             ;EXIT
2740 013126 010337 013140          HRD:  MOV     R3,LOG3B+2    ;HARD ERROR CODE
2741 013132 011137 013142          MOV     @R1,LOG3B+4    ;DESCRIPTION
2742 013136          ERRHRD  0,LOG3B,ERRDES
                                LOG3B:
                                013136 104456
                                013140 000000
                                013142 013136
                                013144 013170
2743
2744 013146 011500          ABO:   MOV     @R5,R0        ;GET UNIT NUMBER
2745 013150 042700 177770          BIC     #177770,R0    ;UN-SIGN EXTEND
2746 013154          DODU     R0             ;USE LOGICAL # TO DROP
```


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

```

2754
2755
2756
2757
2758
2759 013170      BGNMSG  ERRDES      ;ERROR DESCRIPTION      ERRDES::
      013170
2760 013170      PUSH    R0              MOV    R0,-(SP)
      013170      010046
2761 013172      PUSH    R2              MOV    R2,-(SP)
      013172      010246
2762 013174      005002
2763 013176      032715      000020
2764 013202      001401
2765 013204      005202
2766 013206      005046
      013210      153716      003307
      013214      010246
      013216      005046
      013220      156516      000060
      013224      012746      013362
      013230      012746      000004
      013234      010600
      013236      104414
      013240      062706      000012
2767 013244      016500      000064
2768 013250      016502      000072
2769 013254      005046
      013256      156516      000077
      013262      005046
      013264      150216
      013266      005046
      013270      156516      000100
      013274      010046
      013276      012746      013442
      013302      012746      000005
      013306      010600
      013310      104414
      013312      062706      000014
2770 013316      005765      000074
2771 013322      001414
2772 013324      016546      000074
      013330      012746      013616
      013334      012746      000002
      013340      010600
      013342      104414
      013344      062706      000006
2773 013350      005065      000074
2774 013354      012602
      013354

```

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

```

      2$:
      CLR    R2              ;PRESET TO DATA TYPE
      BIT    #BIT4,R5        ;WHAT PACK TYPE?
      BEQ    2$              ;DATA
      INC    R2              ;COMMAND
      PRINTB #UNIT,<B,DR(R5)>,R2,<B,SYSTAT+1>

      CLR    -(SP)
      BISB   SYSTAT+1,(SP)
      MOV    R2,-(SP)
      CLR    -(SP)
      BISB   DR(R5),(SP)
      MOV    #UNIT,-(SP)
      MOV    #4,-(SP)
      MOV    SP,R0
      TRAP   C$PNTB
      ADD    #12,SP

      MOV    REC(R5),R0      ;RECORD NUMBER
      MOV    PATTEN(R5),R2   ;DATA EXPECTED
      PRINTB #RECID,R0,<B,CMD$NT(R5)>,<B,R2>,<B,SUCCS+1(R5)>

      CLR    -(SP)
      BISB   SUCCS+1(R5),(SP)
      CLR    -(SP)
      BISB   R2,(SP)
      CLR    -(SP)
      BISB   CMD$NT(R5),(SP)
      MOV    R0,-(SP)
      MOV    #RECID,-(SP)
      MOV    #5,-(SP)
      MOV    SP,R0
      TRAP   C$PNTB
      ADD    #14,SP

      TST    DLV(R5)         ;DLV ERROR?
      BEQ    3$              ;NO
      PRINTB #RECID2,DLV(R5) ;YES-PRINT

      MOV    DLV(R5),-(SP)
      MOV    #RECID2,-(SP)
      MOV    #2,-(SP)
      MOV    SP,R0
      TRAP   C$PNTB
      ADD    #6,SP

      3$:
      CLR    DLV(R5)         ;RESET
      POP    R2              ;RESTORE
      MOV    (SP)+,R2

```

```

2775 013356          POP      R0
      013356 012600          MOV      (SP)+,R0
2776 013360          ENDMSG          ;EXIT
      013360          L10003:
      013360 104423          TRAP      C$MSG
2777 013362          045      101      104  UNIT:: .ASCIZ  /%ADRIE# %01%A  PAK SENT %01%A  FLAG RCVD %03%N/
2778          .EVEN
2779 013442          045      101      102  RECID:: .ASCIZ  /%ABLOCK# %04%A  COMMAND %02%A  EXPCTD %03%A  SUCCESS %03%N/
2780          .EVEN
2781 013534          103      101      116  OVRFLO: .ASCIZ  /CAN'T UPDATE ERROR OR STATISTIC:OVERFLOW PENDING/
2782          .EVEN
2783 013616          045      101      040  RECID2: .ASCIZ  /%A  RCDB WAS %06%N/
2784          .EVEN

```

```
2787      .SBTTL WHCHDR / SEE WHICH DRIVE IS ACTIVE
2788
2789      :++
2790      : INPUTS: DR(R5)
2791      : OUTPUTS: CARRY=DRIVE (1 OR 0)
2792      :--
2793
2794
2795 013640 000241      WHCHDR:: CLC                      ;CLEAR CARRY
2796
2797 013642 105765 000060      TSTB DR(R5)                ;DR 0?
2798 013646 001401      BEQ 2$                          ;YES
2799 013650 000261      SEC                              ;NO
2800
2801 013652 000207      2$: RETURN                      ;RETURN
```

```

2804 .SBTTL CHKSUM / FORM THE PACKET CHECKSUM
2805
2806 :++
2807 : THE CHECKSUM IS A 16 BIT CHECKSUM WITH END-AROUND CARRY.
2808 :
2809 : INPUTS: R0 -> (POINTS TO) TOP OF PACKET
2810 :         R1 = # OF BYTES
2811 : OUTPUTS: R0 -> WHERE TO PUT CHECKSUM
2812 :         R1 = CHECKSUM
2813 :--
2814
2815
2816 013654 010346 CHKSUM:: PUSH R3 MOV R3,-(SP)
013654
2817 013656 010246 PUSH R2 MOV R2,-(SP)
013656
2818 013660 042737 000001 003306 BIC #BIT0,SYSTAT ;'CHECKSUM IS ODD' BIT
2819 013666 032701 000001 BIT #BIT0,R1 ;AN ODD # OF BYTES?
2820 013672 001403 BEQ 1$ ;NO
2821 013674 052737 000001 003306 BIS #BIT0,SYSTAT ;YES
2822
2823 013702 006001 1$: ROR R1 ;/2 FOR WORDS
2824
2825 013704 005003 2$: CLR R3 ;PREP CHECKSUM WORD
2826
2827 013706 062003 3$: ADD (R0)+,R3 ;FORM SUM
2828 013710 005503 ADC R3 ;WITH CARRY
2829 013712 005301 DEC R1 ;MORE WORDS?
2830 013714 001374 BNE 3$ ;YES
2831
2832 013716 032737 000001 003306 BIT #BIT0,SYSTAT ;WAS IT ODD
2833 013724 001405 BEQ 4$ ;NO
2834 013726 112002 MOV8 (R0)+,R2 ;YES GET NEXT BYTE
2835 013730 042702 177400 PIC #177400,R2 ;UN-SIGN EXTEND
2836 013734 060203 ADD R2,R3 ;ADD IT IN
2837 013736 005503 ADC R3 ;AND CARRY JUST IN CASE
2838
2839 013740 010301 4$: MOV R3,R1 ;RETURN IT IN CORRECT PLACE
2840 013742 012602 POP R2 ;RESTORE
013742 MOV (SP)+,R2
2841 013744 012603 POP R3 MOV (SP)+,R3
013744
2842 013746 000207 RETURN ;RETURN

```

```

2845      .SBTTL  CKCKSM / MODULE TO CHECK THE CHKSUMS
2846
2847      :++
2848      : MAKE SURE THE CHECKSUM RECEIVED = THE CHECKSUM CALCULATED.
2849      : INPUTS:  R4 = THE PACKET BYTE COUNT
2850      :           R0 -> THE PACKET TOP
2851      : OUTPUTS:  CARRY SET IF CHECKSUM CALC'D DOES NOT EQUAL CHECKSUM SENT
2852      :           R0 -> THE PACKET TOP
2853      :--
2854
2855
2856 013750      CKCKSM:: PUSH  R1
013750                                MOV    R1,-(SP)

2857 013752      PUSH  R0
013752                                ;SAVE
MOV    R0,-(SP)

2858 013754      MOV    R4,R1
013756 010401      CALL  CHKSUM      ;COPY BYTE COUNT TO CORRECT
013654                                ;REGISTER FOR CHKSUM AND
;FORM CHECKSUM

2860      ;HERE R0 --> XMITTED CHKSUM, R1-CHKSUM CALC'D
2861
2862
2863
2864 013762      CMPB   (R0)+,R1      ;LOWER ORDER CHECK
2865 013764      BNE    2$            ;WRONG
2866
2867 013766      SWAB   R1            ;OK-PREP FOR
2868
2869 013770      CMPB   (R0)+,R1      ;HIGH ORDER CHECK
2870 013772      BNE    2$            ;WRONG
2871 013774      CLC                     ;OK-CLEAR SAILING
2872
2873 013776      BR     3$            ;EXIT
2874
2875 014000      2$:   SEC                     ;LET ERROR BE KNOWN
2876
2877
2878 014002      3$:   POP    R0
014002                                MOV    (SP)+,R0

2879 014004      POP    R1
014004                                MOV    (SP)+,R1

2880 014006      RETURN
                                ;RETURN

```



```

2883 .SBTTL DOBRK / MODULE TO INIT TU58 AND TEST INTERRUPTS
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905 014010 105037 014543 DOBRK:: CLRB INITWD+1 ;CLEAR BYTE RECEIVE ADDR
2906 014014 C05037 014544 CLR BRKTO ;CLEAR TIME OUT CONSTANT
2907 014020 052775 000001 000026 BIS #BIT0,@XMSR(R5) ;SET 'BREAK'
2908 014026 012765 000001 000100 MOV #RSSNIT,CMSNT(R5) ;SAY WE SENT 'INIT'
2909 014034 052715 000020 BIS #BIT4,@R5 ;PAK SENT TYPE =COMMAND, SORT OF
2910 014040 012704 000010 MOV #8.,R4 ;BREAK-IT'S-BACK COUNT=8
2911 014044 104422 1$: BREAK ;SUPERVISOR TAKE FIVE TRAP CSBRK
2912 014046 105775 000026 ;FOR ^C CHECK, ETC.
2913 014052 100410 TSTB @XMSR(R5) ;READY?
2914 014054 005337 014544 BMI 4$ ;YES
2915 014060 001371 DEC BRKTO ;NO, TIME OUT?
2916 014062 012704 000056 BNE 1$ ;NO
2917 014066 004737 012634 MOV #TOSNDB,R4 ;YES, SET ERROR CODE
2918 014072 000535 CALL LOG ;LOG IT
2919 014074 113775 014540 000030 4$: MOVBR BRKWD,@XMSR(R5) ;SEND NULL
2920 014102 005037 014544 CLR BRKTO ;RESET TIME OUT
2921 014106 005304 DEC R4 ;MORE NULLS TO SEND?
2922 014110 001355 BNE 1$ ;YES
2923 014112 005375 000026 DEC @XMSR(R5) ;NO, CLEAR 'BREAK'
2924 014116 017500 000024 MOV @RCDB(R5),R0 ;HEAVE 'GARBAGE' 1ST BYTE
2925 014122 012700 000000 SETPRI #PRI00 ;SET TO INTERRUPT FO SURE
2926 014126 104441 MOV #PRI00,R0
2927 014130 012746 000340 TRAP CSSPRI
2928 014134 012746 014450 MOV #PRI07,-(SP)
2929 014140 016546 000204 MOV #RCVINT,-(SP)
2930 014144 012746 000003 TUVECT(R5),-(SP)
2931 014150 104437 MOV #3,-(SP)
2932 014152 062706 000010 TRAP CSSVEC
2933 014156 062765 000004 000204 ADD #4,TUVECT(R5) ;AND INC TO SND VECTOR
2934 014164 012746 000340 SETVEC TUVECT(R5),#SNDINT,#PRI07;AND SET IT MOV #PRI07,-(SP)

```

014170	012746	014434					MOV	#SNDINT,-(SP)
014174	016546	000204					MOV	TUVECT(R5),-(SP)
014200	012746	000003					MOV	#3,-(SP)
014204	104437						TRAP	C\$SVEC
014206	062706	000010					ADD	#10,SP
2930	014212	162765	000004	000204		SUB	#4,TUVECT(R5)	;RESET VECTOR ADDR.
2931	014220	005037	014544			CLR	BRKTO	;RESET TIME OUT
2932	014224	012704	014542			MOV	#INITWD,R4	;USE ADDR. FOR SNDBYT
2933	014230	010437	014546			MOV	R4,BRKPTR	;AND SAVE FOR 'WAIT'
2934	014234	052775	000100	000026		BIS	#BIT6,@XMSR(R5)	;ENABLE INTER.
2935	014242	004737	014504			CALL	WAIT	;AND ENTER LOOP
2936	014246	005715				TST	@R5	;ABORTED FROM TIME OUT?
2937	014250	100446				BMI	3\$	;YES-EXIT
2938								
2939	014252	005037	014544			CLR	BRKTO	;RESET TIME OUT
2940	014256	012704	014542			MOV	#INITWD,R4	;SEND SECOND INIT
2941	014262	010437	014546			MOV	R4,BRKPTR	;SAVE POINTER AGAIN
2942	014266	052775	000100	000026		BIS	#BIT6,@XMSR(R5)	;AND THEN ENABLE INT
2943	014274	004737	014504			CALL	WAIT	;AND WAIT
2944	014300	005715				TST	@R5	;IF ABORTED
2945	014302	100431				BMI	3\$	;THEN EXIT
2946								
2947	014304	012704	014543			MOV	#INITWD+1,R4	;WHERE RESPONSE WILL GO (ADDRESS)
2948	014310	010437	014546			MOV	R4,BRKPTR	;AND FOR 'WAIT'
2949	014314	052775	000100	000022		BIS	#BIT6,@RCSR(R5)	;ENABLE RECIEVE INT.
2950	014322	004737	014504			CALL	WAIT	;GET ANSWER
2951	014326	005715				TST	@R5	;ABORTED?
2952	014330	100416				BMI	3\$	;YES.
2953								
2954	014332	123727	014543	000020		CMPB	INITWD+1,#RSCONT	;NO, IS IT 'CONTINUE'?
2955	014340	001003				BNE	2\$	;NOPE-ERROR
2956								
2957	014342	042715	040000			BIC	#BIT14,@R5	;SUCCESSFUL, CLEAR DOBREAK FLAG
2958	014346	000407				BR	3\$	;EXIT
2959								
2960	014350	113737	014543	003307	2\$:	MOVB	INITWD+1,SYSTAT+1	;SAVE BUM RESPONSE
2961	014356	012704	000032			MOV	#CNINIT,R4	;CAN'T INIT CODE
2962	014362	004737	012634			CALL	LOG	;LOG IT
2963								;SCHEDULER WILL TRY AGAIN IF NOT ABORTED
2964								
2965	014366	042775	000100	000026	3\$:	BIC	#BIT6,@XMSR(R5)	;CLEAR INTERRUPTS
2966	014374	042775	000100	000022		BIC	#BIT6,@RCSR(R5)	;AND FOR RECIEVE
2967	014402					CLRVEC	TUVECT(R5)	;RELEASE RECIEVE VECT.
	014402	016500	000204					
	014406	104436						
2968	014410	062765	000004	000204		ADD	#4,TUVECT(R5)	;AND GET SEND ADDR.
2969	014416					CLRVEC	TUVECT(R5)	;AND RELEASE IT
	014416	016500	000204					
	014422	104436						
2970	014424	162765	000004	000204		SUB	#4,TUVECT(R5)	;RESTORE POINTER
2971	014432	000207				RETURN		;RETURN


```

2974 .SBTTL INTERRUPT SERVICE ROUTINES AND TIMER
2975
2976 014434 BGNSRV SNDINT ;'SEND' INTERRUPT SERVICE:
      014434                                SNDINT::
2977
2978 014434 042775 000100 000026 SNDHND: BIC #BIT6,@XMSR(R5) ;DISABLE INTERRUPT
2979 014442 112475 000030          MOV (R4)+,@XMDB(R5);OUTPUT BYTE
2980 014446 ENDSRV
      014446                                L10004:
      014446 000002                                RTI
2981
2982
2983
2984 014450 BGNSRV RCVINT ;'RCV' INTERRUPT SERVICE:
      014450                                RCVINT::
2985
2986 014450 042775 000100 000022 RCVHND: BIC #BIT6,@RCR(R5) ;DISABLE INTS
2987 014456 017565 000024 000074          MOV @RCDB(R5),DLV(R5) ;SAVE WORD
2988 014464 116524 000074          MOV DLV(R5),(R4)+ ;BYTE TO BUFFER
2989 014470 005765 000074          TST DLV(R5) ;ERROR?
2990 014474 100402          BMI 10$ ;YES
2991 014476 005065 000074          CLR DLV(R5) ;NO CLEAR ERROR
2992 014502 10$:
2993 014502 ENDSRV                                L10005:
      014502 000002                                RTI
2994
2995
2996
2997 014504 000240 WAIT: NOP ;WAIT LOOP FOR
2998                                ;INTERRUPT SERVICING
2999 014506 020437 014546          CMP R4,BRKPTR ;IF=,THEN NO INTERRUPT
3000 014512 001011          BNE 1$ ;GOT ONE!
3001 014514          BREAK ;SUPERVISOR BREAK
      014514 104422                                TRAP CSBRK
3002 014516          BREAK ;KILL SOME TIME                                TRAP CSBRK
      014516 104422
3003 014520 005337 014544          DEC BRKTO ;TIME OUT?
3004 014524 001367          BNE WAIT ;NO...CONT.
3005 014526 012704 000050          MOV #TORCVB,R4 ;YES LOAD ERROR #
3006 014532 004737 012634          CALL LOG ;LOG IT
3007 014536 000207 1$:          RETURN ;RETURN
3008
3009 014540 000000 BRKWD: .WORD 0 ;NULL
3010 014542 004          INITWD: .BYTE RSINIT ;INIT COMMAND
3011 014543 000          .BYTE 0 ;RSCONT IS EXPECTED HERE
3012 014544 000000 BRKTO: .WORD 0 ;TIME OUT
3013 014546 000000 BRKPTR: .WORD 0 ;POINTER TO INITWD

```

```
3016 .SBTTL COMPAR/DATA COMPARISON MODULE
3017
3018
3019 :++
3020 : COMPAR - IF "COMPARE DATA" SELECTED, COMPARE EACH DATA BYTE OF PACKET
3021 :           TO PATTEN(R5).  SAVE NUMBER OF BYTES NOT CORRECT.  IF NOT
3022 :           0, PRINT SOFT ERROR AND TOTAL # WRONG BYTES.  SET 'BAD_DATA_
3023 :           IN_PACKET' BIT (BIT6 @R5) FOR HIGHER LEVEL MODULES.
3024 : INPUTS:  - (CMPDAT) FLAG TO NOT COMPARE (=1)
3025 :           - PKPTR(R5) POINTS TO DATA PACK.
3026 : OUTPUTS: BIT6 @R5 (BAD DATA FLAG) ADJUSTED.
3027 :           L$UN - UNIT NUMBER
3028 :           PRNSIZ - SIZE OF PACKET
3029 :--
3030 COMPAR:: PUSH    R0                ;COMPARE DATA IS DATA PACKET
      014550      MOV     R0,-(SP)
      014550      010046
3031      014552      PUSH    R4                ;TO PATTERN WRITTEN
      014552      010446      MOV     R4,-(SP)
3032      014554      PUSH    R1                ;USING BYTE COUNT IN PACKET
      014554      010146      MOV     R1,-(SP)
3033 014556 005037 014726      CLR     BDBYTS      ;CLEAR TOTAL WRONG
3034 014562 016504 000104      MOV     PKPTR(R5),R4 ;GET TOP OF PACKET
3035 014566 005737 002216      TST     CMPDAT      ;COMPARE SELECTED?
3036 014572 001451      BEQ     4$      ;NO-EXIT
3037 014574 005204      INC     R4      ;YES, LOCATE COUNT
3038 014576 111401      MOVB    @R4,R1      ;GET IT
3039 014600 042701 177400      BIC     #177400,R1 ;SIGN-UNEXTEND
3040      ;MUST TEST BYTE-WISE...
3041 014604 005204      INC     R4      ;-->FIRST DATA BYTE
3042 014606 126524 000072 1$: CMPB    PATTEN(R5),(R4)+ ;DATA-WHAT WAS EXPECTED?
3043 014612 001402      BEQ     2$      ;YES
3044 014614 005237 014726      INC     BDBYTS      ;NO, INCREMENT TOTAL WRONG
3045 014620 005301 2$: DEC     R1      ;MORE LEFT?
3046 014622 001371      BNE     1$      ;YES
3047 014624 005737 014726      TST     BDBYTS      ;ANY WRONG?
3048 014630 001432      BEQ     4$      ;NO
3049 014632 011537 002074      MOV     @R5,L$UN      ;GET UNIT NUMBER
3050 014636 042737 177770 002074      BIC     #177770,L$UN ;MASK IT OFF
3051 014644      ERRSOFT 0.,MSBDA,ERRDES ;YES-PRINT 'BAD DATA IN PACKET' ERROR
      014644      104457      TRAP    CSERSOFT
      014646      000000      .WORD   0
      014650      002340      .WORD   MSBDA
      014652      013170      .WORD   ERRDES
3052 014654      PRINTB  #DESC,BDBYTS
      014654      013746 014726      MOV     BDBYTS,-(SP)
      014660      012746 014730      MOV     #DESC,-(SP)
      014664      012746 000002      MOV     #2,-(SP)
      014670      010600      MOV     SP,R0
      014672      104414      TRAP    C$PNTB
      014674      062706 000006      ADD     #5,SP
3053 014700 052715 000100      BIS     #BIT6,@R5 ;LET 'EM KNOW UPSTAIRS-BAD DATA FLAG
```

```

3054 014704 012737 000204 003336      MOV    #132.,PRNSIZ ;SIZE IS ONE DATA PACK
3055 014712 004737 014764      CALL   PRNPAK    ;AND PRINT THE PACKET
3056 014716      4$:      POP     R1      ;RESTORE
      014716 012601      MOV     (SP)+,R1
3057 014720      POP     R4      MOV     (SP)+,R4
      014720 012604
3058 014722      POP     R0      MOV     (SP)+,R0
      014722 012600
3059
3060 014724 000207      RETURN
3061
3062 014726 000000      BDBYTS: .WORD
3063 014730 045      101      124  DESC: .ASCIZ  /%ATOTAL BAD BYTES= %D3%A.%N/
3064      .EVEN

```

```

3067 .SBTTL PRNPAK/MODULE TO PRINT DATA PACKET
3068
3069 :++
3070 : PRNPAK - IF PRINT DATA PACK_ON_ERROR SELECTED: PRINT EACH BYTE OF PACKET
3071 : TO BY PKPTR(R5).
3072 : INPUTS: PRNSIZ - # OF BYTES IN PACKET.
3073 : OUTPUTS: NONE
3074 :--
3075
3076 014764 000240 PRNPAK:: NOP ;PRINTS 1 PACKET
3077 ;PKPTR(R5)->TOP OF PACKET
3078 ;PRNSIZ (PASSED)=BYTE COUNT
3079 014766 010046 PUSH R0 MOV R0,-(SP)

3080 014770 010446 PUSH R4 MOV R4,-(SP)

3081 014772 105737 002214 TSTB PRBUF ;PRINT PACKET SELECTED?
3082 014776 001451 BEQ 4$ ;NO
3083 015000 016504 000104 MOV PKPTR(R5),R4 ;YES-GET TOP OF PACK
3084 015004 012737 000020 015130 1$: MOV #16,,LNCNT ;16 BYTES PER LINE
3085 015012 112437 015132 2$: MOVB (R4)+,PRDAT ;AVOID SIGN EXTEND
3086 015016 PRINTF #PRFORM,<B,PRDAT> ;PRINT BYTE
015016 005046 CLR -(SP)
015020 153716 015132 BISB PRDAT,(SP)
015024 012746 015134 MOV #PRFORM,-(SP)
015030 012746 000002 MOV #2,-(SP)
015034 010600 MOV SP,R0
015036 104417 TRAP C$PNTF
015040 062706 000006 ADD #6,SP
3087 015044 005337 003336 DEC PRNSIZ ;ONE LESS
3088 015050 001414 BEQ 3$ ;NO MORE
3089 015052 005337 015130 DEC LNCNT ;NEW LINE?
3090 015056 001355 BNE 2$ ;NOT YET
3091 015060 PRINTF #CARLF ;YES
015060 012746 015144 MOV #CARLF,-(SP)
015064 012746 000001 MOV #1,-(SP)
015070 010600 MOV SP,R0
015072 104417 TRAP C$PNTF
015074 062706 000004 ADD #4,SP
3092 015100 000741 BR 1$ ;NEXT LINE
3093 015102 012746 015144 3$: PRINTF #CARLF ;FINISH UP
015106 012746 000001 MOV #CARLF,-(SP)
015112 010600 MOV #1,-(SP)
015114 104417 MOV SP,R0
015116 062706 000004 TRAP C$PNTF
3094 015122 012604 4$: POP R4 MOV (SP)+,R4
3095 015124 012600 POP R0 MOV (SP)+,R0
3096 015126 000207 RETURN ;RETURN

```

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

3119 .TITLE MISCELLANEOUS SECTIONS
3120 .SBTTL REPORT CODING SECTION
3148
3149 015150 BGNMOD
3150
3151 :++
3152 : THE REPORT CODING SECTION CONTAINS THE
3153 : 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
3154 :--
3155

3156 015150 BGNRPT
015150 LSRPT::
3157 015150 010046 PUSH R0 MOV R0,-(SP)
015150

3158 015152 010146 PUSH R1 MOV R1,-(SP)
015152

3159 015154 010246 PUSH R2 MOV R2,-(SP)
015154

3160 015156 010346 PUSH R3 MOV R3,-(SP)
015156

3161 015160 010446 PUSH R4 MOV R4,-(SP)
015160

3162 015162 010546 PUSH R5 MOV R5,-(SP)
015162

3163
3164 015164 104422 003350 015576 BREAK TRAP C\$BRK
015164 012737
3165 015166 012737 003350 015576 MOV #BLKTBL,RPTR ;GET 1ST DEVICE BLOCK
3166 015174 012746 015600 PRINTS #STATHD ;HEADER
015174 012746 000001 MOV #STATHD,-(SP)
015200 012746 000001 MOV #1,-(SP)
015204 010600 MOV SP,R0
015206 104416 TRAP C\$PNTS
015210 062706 000004 ADD #4,SP
3167 015214 104422 BREAK ;^C CHECK
015214 104422 TRAP C\$BRK
3168 015216 012746 016054 PRINTS #STHD2 ;2ND HEADER
015216 012746 000001 MOV #STHD2,-(SP)
015222 012746 000001 MOV #1,-(SP)
015226 010600 MOV SP,R0
015230 104416 TRAP C\$PNTS
015232 062706 000004 ADD #4,SP
3169 015236 104422 1\$: BREAK ;^C CHECK
015236 017705 000332 MOV @RPTR,R5 TRAP C\$BRK
3170 015240 017705 000332 ;GET DEVICE BLOCK

MISCELLANEOUS SECTIONS
REPORT CODING SECTION

MACRO M1113 25-FEB-82 07:06 PAGE 89-1

F 7

SEQ 0083

3171	015244	032715	004000	BIT	#BIT11,R5	:UNIT NOT TESTED?	
3172	015250	001131		BNE	2\$	:TRUE, DON'T PRINT STATISTICS	
3173						:OK TO PRINT	
3174	015252	011537	015574	MOV	R5,RLUN	:SAVE STATUS WORD	
3175	015256	042737	177770	BIC	#177770,RLUN	:MASK UNIT NUM.	
3176	015264	116501	000122	MOVB	SOFTW(R5),R1	:SOFTREAD	
3177	015270	042701	177400	BIC	#177400,R1	:SIGN-UNEXTEND	
3178	015274	116502	000124	MOVB	SOFTW(R5),R2	:SOFT WRITE	
3179	015300	042702	177400	BIC	#177400,R2		
3180	015304	116503	000136	MOVB	HARDR(R5),R3	:HARD READ	
3181	015310	042703	177400	BIC	#177400,R3		
3182	015314	116504	000140	MOVB	HARDW(R5),R4	:HARD WRITE	
3183	015320	042704	177400	BIC	#177400,R4		
3184	015324			PRINTS	#FM0,RLUN	:SUMMARY/UNIT #	
	015324	013746	015574				MOV RLUN,-(SP)
	015330	012746	015712				MOV #FM0,-(SP)
	015334	012746	000002				MOV #2,-(SP)
	015340	010600					MOV SP,R0
	015342	104416					TRAP C\$PNTS
	015344	062706	000006				ADD #6,SP
3185	015350			PRINTS	#FM,#0,WRTN0(R5),RDN0(R5),<B,BDATA(R5)>,R1,R2,R3,R4		
	015350	010446					MOV R4,-(SP)
	015352	010346					MOV R3,-(SP)
	015354	010246					MOV R2,-(SP)
	015356	010146					MOV R1,-(SP)
	015360	005046					CLR -(SP)
	015362	156516	000134				BISB BDATA(R5),(SP)
	015366	016546	000114				MOV RDN0(R5),-(SP)
	015372	016546	000110				MOV WRTN0(R5),-(SP)
	015376	012746	000000				MOV #0,-(SP)
	015402	012746	015730				MOV #FM,-(SP)
	015406	012746	000011				MOV #11,-(SP)
	015412	010600					MOV SP,R0
	015414	104416					TRAP C\$PNTS
	015416	062706	000024				ADD #24,SP
3186	015422	116501	000123	MOVB	SOFTW+1(R5),R1	:SAME	
3187	015426	042701	177400	BIC	#177400,R1	:AS	
3188	015432	116502	000125	MOVB	SOFTW+1(R5),R2	:ABOVE	
3189	015436	042702	177400	BIC	#177400,R2	:THIS	
3190	015442	116503	000137	MOVB	HARDR+1(R5),R3	:TIME	
3191	015446	042703	177400	BIC	#177400,R3	:FOR	
3192	015452	116504	000141	MOVB	HARDW+1(R5),R4	:DRIVE	
3193	015456	042704	177400	BIC	#177400,R4	:ONE	
3194							
3195	015462			PRINTS	#FM,#1,WRTN1(R5),RDN1(R5),<B,BDATA+1(R5)>,R1,R2,R3,R4		
	015462	010446					MOV R4,-(SP)
	015464	010346					MOV R3,-(SP)
	015466	010246					MOV R2,-(SP)
	015470	010146					MOV R1,-(SP)
	015472	005046					CLR -(SP)
	015474	156516	000135				BISB BDATA+1(R5),(SP)
	015500	016546	000116				MOV RDN1(R5),-(SP)
	015504	016546	000112				MOV WRTN1(R5),-(SP)
	015510	012746	000001				MOV #1,-(SP)
	015514	012746	015730				MOV #FM,-(SP)
	015520	012746	000011				MOV #11,-(SP)
	015524	010600					MOV SP,R0

MISCELLANEOUS SECTIONS
REPORT CODING SECTION

MACRO M1113 25-FEB-82 07:06 PAGE 89-2

G 7

SEQ 0084

									TRAP ADD	C\$PNTS #24,SP
	015526	104416								
	015530	062706	000024							
3196	015534	023727	015576	003366	2\$:	CMP	RPTR,#LSTDEV	:ALL UNITS DONE?		
3197	015542	103005				BHIS	3\$	:YES		
3198	015544	062737	000002	015576		ADD	#2,RPTR	:NO-DO		
3199										
3200	015552	000137	015236			JMP	1\$	:MORE UNITS		
3201										
3202	015556				3\$:	POP	R5			
	015556	012605						MOV	(SP)+,R5	
3203	015560					POP	R4			
	015560	012604						MOV	(SP)+,R4	
3204	015562					POP	R3			
	015562	012603						MOV	(SP)+,R3	
3205	015564					POP	R2			
	015564	012602						MOV	(SP)+,R2	
3206	015566					POP	R1			
	015566	012601						MOV	(SP)+,R1	
3207	015570					POP	R0			
	015570	012600						MOV	(SP)+,R0	
3208	015572					ENDRPT				
	015572									
	015572	104425								
3209	015574	000000				RLUN:	.WORD			
3210	015576	000000				RPTR:	.WORD			
3211										
3212	015600	045	116	045	STATHD:	.ASCII	/XN% A DR BLKS WR BLKS RD BDPK /			
3213	015646	104	103	110		.ASCII	@DCHK/RD DCHK/WR DCHK/RD DCHK/WRXN@			
3214						.EVEN				
3215	015712	045	101	125	FM0:	.ASCII	/XAUNIT XD1XN/			
3216						.EVEN				
3217										
3218	015730	045	101	040	FM:	.ASCII	/XA XD1XA XD5XA. XD5XA. XD3XA. /			
3219	016004	045	104	063		.ASCII	/XD3XA. XD3XA. XD3XA. XD3XA.XN/			
3220						.EVEN				
3221	016054	045	101	040	STHD2:	.ASCII	/XA			
3222	016121	122	105	103		.ASCII	/RECOV RECOV UNRECOV UNRECOVXN/			
3223						.EVEN				
3224	016164					ENDMOD				

L10006: TRAP C\$RPT


```

3227          .SBTTL  INITIALIZE SECTION
3228
3229          :++
3230          : THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3231          : AT THE BEGINNING OF EACH PASS.
3232          :--
3233
3234 016164          BGNINIT
3235 016164          LSINIT::
3236 016164 000240          INIT: NOP
3240 016166 105037 016746          CLR      STRT      ;FOR STATS CLEAR
3241 016172 005037 003342          CLR      TEST9    ;***** CLR TST 9 FLAG
3242 016176          READEF  #EF.START    ;START COMMAND?
3243 016176 012700 000040          MOV      #EF.START,R0
3244 016202 104447          TRAP      CSREFG
3245 016204          BNCOMPLETE INIT2      ;NO
3246 016204 103002          BCC      INIT2
3247 016206 005237 016746          INC      STRT      ;YES, SET START FLAG
3248 016212 012737 003350 003312  INIT2: MOV      #BLKTBL,DEVPTN ;SET ALL UNITS ABORTED:
3249 016220 005004          CLR      R4            ;UNIT NUMBER
3250 016222 017705 165064          MOV      @DEVPTN,R5 ;GET POINTER
3251 016226 010415          MOV      R4,@R5        ;INSERT UNIT #
3252 016230 052715 100000          BIS      #BIT15,@R5 ;SET ABORTED
3253 016234 052715 004000          BIS      #BIT11,@R5 ;SET UNIT NOT TESTED
3254 016240 006304          ASL      R4            ;*2 FOR LOOK-UP
3255 016242 016465 027700 000102  MOV      BUFTBL(R4),RCVBUF(R5) ;SETUP POINTER TO UNIT'S BUFFER
3256 016250 006204          ASR      R4            ;CORRECT BACK TO UNIT #
3257 016252 023727 003312 003366  CMP      DEVPTN,#LSTDEV ;LAST DEVICE DONE?
3258 016260 103005          BHS      CHECK        ;YES
3259 016262 062737 000002 003312  ADD      #2,DEVPTN ;NO-GET
3260 016270 005204          INC      R4            ;NEXT DEVICE AND
3261 016272 000753          BR       1$           ;SERVICE
3262 016274 022737 000010 002012  CHECK: CMP      #8,LSUNIT ;MAKE SURE NOT
3263 016302 103005          BHS      GETHRD      ;TOO MANY UNITS
3264 016304          ERRSF  101.,TOMANY ;TOMANY-REQUEST ^C
3265 016304 104454          TRAP      CSERSF
3266 016306 000145          .WORD    101
3267 016310 016664          .WORD    TOMANY
3268 016312 000000          .WORD    0
3269 016314          DOCLN          ;EXIT
3270 016314 104444          TRAP      CSDCLN
3271 016316 012737 003350 003312  GETHRD: MOV      #BLKTBL,DEVPTN ;INIT TABLE POINTER
3272 016324 005004          CLR      R4            ;CLEAR DEVICE COUNTER
3273 016326 017705 164760          MOV      @DEVPTN,R5 ;GET STATUS WORD
3274 016332 010437 002074          MOV      R4,LSLUN ;UNIT NUM. IN CASE ERROR
3275 016336          GPWARD  R4,R2            ;GET HARD INFO
3276 016336 010400          MOV      R4,R0
3277 016340 104442          TRAP      C$GPWARD
3278 016342 010002          MOV      R0,R2
3279 016344          BNCOMPLETE 3$
3280 016344 103111          BCC      3$
3281 016346 042715 004000          BIC      #BIT11,@R5 ;UNIT IS TESTED!
3282 016352 012203          MOV      (R2)+,R3      ;R3=CSR
3283 016354 012265 000204          MOV      (R2)+,TUVECT(R5) ;GET VECTOR ADDRESS

```

MISCELLANEOUS SECTIONS
INITIALIZE SECTION

MACRO M1113 25-FEB-82 07:06 PAGE 91-1

I 7

SEQ 0086

3274	016360	112265	000061		MOVB	(R2)+,DR+1(R5)	:SAVE UNIT SUMMARY		
3275	016364	005202			INC	R2	:GET TO WORD BOUND		
3276	016366	012237	016750		MOV	(R2)+,PDTFLG	:AND GET PDT FLAG		
3277	016372	052715	040000		BIS	#BIT14,R5	:SET SEND BREAK FLAG		
3278	016376	052765	000400	000060	BIT	#BIT8,DR(R5)	:DRIVE 0?		
3279	016404	001011			BNE	13\$	:YES		
3280	016406	032765	001000	000060	BIT	#BIT9,DR(R5)	:DRIVE 1?		
3281	016414	001005			BNE	13\$	:OK		
3282	016416				ERRSF	102.,NODRVS	:NEITHER?!		
	016416	104454						TRAP	C\$ERSF
	016420	000146						.WORD	102
	016422	016714						.WORD	NODRVS
	016424	000000						.WORD	0
3283	016426				DOCLN		:EXIT	TRAP	C\$DCLN
	016426	104444							
3284									
3285	016430	105737	016746	13\$:	TSTB	STRT	:START COMMAND?		
3286	016434	001412			BEQ	14\$	:NO, DONT CLEAR		
3287							:YES-CLEAR STATS		
3288	016436	012702	000202		MOV	#BLKEND,R2	:R2-->END OF STATS		
3289	016442	012701	000110		MOV	#WRTNO,R1	:FORM ADDRESS OF START		
3290	016446	060501			ADD	R5,R1	:R1-->START OF STATS.		
3291	016450	162702	000110		SUB	#WRTNO,R2	:FORM # TO CLEAR		
3292									
3293	016454	105021		2\$:	CLRB	(R1)+	:CLEAR 'EM		
3294	016456	005302			DEC	R2	:MORE?		
3295	016460	001375			BNE	2\$	:YES		
3296	016462	042715	100000	14\$:	BIC	#BIT15,R5	:SET NOT ABORTED		
3297	016466	010365	000022		MOV	R3,RCR(R5)	:GET DEVICE REGISTERS:		
3298	016472	062703	000002		ADD	#2,R3			
3299	016476	010365	000024		MOV	R3,RCDB(R5)			
3300	016502	062703	000002		ADD	#2,R3			
3301	016506	010365	000026		MOV	R3,XMSR(R5)			
3302	016512	062703	000002		ADD	#2,R3			
3303	016516	105737	016750		TSTB	PDTFLG	:UNIT A PDT?		
3304	016522	001402			BEQ	4\$	:NO		
3305	016524	162703	000004		SUB	#4,R3	:YES...RCDB=XMDB		
3306	016530	010365	000030	4\$:	MOV	R3,XMDB(R5)			
3307	016534	005065	000072		CLR	PATTEN(R5)	:ZERO DATA PATTERN		
3308	016540	005065	000002		CLR	RETRY(R5)	:NO RETRIES		
3309	016544	005065	000064		CLR	REC(R5)	:NO RECORD		
3310	016550	005065	000076		CLR	SUCCS(R5)	:NO SUCCESS		
3311	016554	005065	000074		CLR	DLV(R5)	:NO DLV ERROR		
3312	016560	005065	000210		CLR	MRSP(R5)	:***** CLR MRSP INDICATOR		
3313	016564	005037	003340		CLR	ALLGON	:OK TO PRINT STATISTICS		
3314	016570	062737	000002	003312 3\$:	ADD	#2,DEVPTN	:-->NEXT DEVICE		
3315	016576	005204			INC	R4	:INCREMENT UNIT NUMBER		
3316	016600	020437	002012		CMP	R4,LSUNIT	:MORE UNITS?		
3317	016604	001250			BNE	1\$	:YES, GP HARD THE NEXT		
3318									
3319	016606	005037	003306		CLR	SYSTAT	:SYSTEM STATUS WORD		
3320	016612				RFLAGS	FLGLOC	:GET USER FLAGS	TRAP	C\$RFLA
	016612	104421						MOV	RO,FLGLOC
	016614	010037	016752						
3321	016620	005037	003332	5\$:	CLR	BLKER	:NO ERROR		
3322	016624	013737	002210	003310	SETLEN:	MOV	LENGTH,TAPLEN	:GET # OF RECORDS	
3323	016632	006237	003310		ASR	TAPLEN	:GET # BLOCKS PER TRACK		

```

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

```

L10007: TRAP CSINIT

```

3362      ;++
3363      ; THE AUTO DROP CODE IS INVOKED WHEN THE ADR FLAG IS SET AND CHECKS FOR
3364      ; A VALID INTERFACE LOCATION. DROPS UNIT IF INTERFACE IS NOT THERE.
3365      ;--
3366
3367 016754      BGNAUTO
3368 016754      000240
3369 016756      NOP      ;AUTO DROP ROUTINE
3370 017004      012746 000340 017062      1$: MOV      #BLKTBL,TRPPTR ;GET TOP OF DATA BLOCK TABLE
3371 017012      017705 000044      MOV      @TRPPTR,R5 ;GET DATA BLOCK
3372 017016      032715 104000      BIT      #B1115!BIT11,@R5 ;NOT TESTED OR ABORTED?
3373 017022      100403      BMI      2$ ;YES
3374 017024      005775 000022      TST      @RCSR(R5) ;NO-VALID ADDRESS?
3375 017030      000240      NOP      ;YES...(TRAP IF NOT)
3376 017032      023727 017062 003366 2$: CMP      TRPPTR,#LSTDEV ;MORE TO TRY?
3377 017040      103004      BHS      3$ ;NO
3378 017042      062737 000002 017062      ADD      #2,TRPPTR ;ON TO NEXT
3379 017050      000760      BR       1$ ;GET IT
3380 017052      3$: CLRVEC #4 ;RESTORE
3381 017052      012700 000004      MOV      #4,R0
3382 017056      104436      TRAP      C$CVEC
3383      ENDAUTO
3384      L10010:
3385      TRAP      C$AUTO
3386      TRPPTR: .WORD
3387
3388      ;ILLEGAL ADDRESS TRAP HANDLER:
3389 017064      TRPHND: PRINTF #MSAUTO ;SAY 'AUTO DROPPED'
3390 017064      012746 017116      MOV      #MSAUTO,-(SP)
3391 017070      012746 000001      MOV      #1,-(SP)
3392 017074      010600      MOV      SP,R0
3393 017076      104417      TRAP      C$PNTF
3394 017100      062706 000004      ADD      #4,SP
3395 017104      011500      MOV      @R5,R0 ;GET UNIT #
3396 017106      042700 177770      BIC      #177770,R0 ;MASK IT OFF
3397 017112      DODU      R0 ;DROP HIM
3398 017112      104451      TRAP      C$DODU
3399 017114      000002      RTI
3400 017116      045      101      101      MSAUTO: .ASCIIZ /%AAUTO DROP: %N/

```

3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3417
3429
3430
017154
017154
017154

017136
017136
017136 005737 003340
017142 001004
017144 005737 002212
017150 001401
017152
017152 104424

017154
017154
017154 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

```

3433                                     .SBTTL  DROP UNIT SECTION
3434
3435                                     :++
3436                                     : THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3437                                     : TO NO LONGER BE TESTED.
3438                                     :--
3439
3440 017156                                BGNDU                                LSDU::
3441 017156
3442 017156 010046                        PUSH    R0                        ;RO=UNIT NUMBER
3443                                     ;SAVE IT
3444                                     MOV      R0,-(SP)
3445
3446 017160 010546                        PUSH    R5                        ;SAVE PRESENT UNIT POINTER
3447                                     MOV      R5,-(SP)
3448
3449 017162 004737 017222                  CALL    GETR5                    ;GET POINTER TO UNIT
3450 017166 052715 100000                  BIS     #BIT15,R5                ;SET ABORTED
3451 017172 012605                        POP     R5                        ;RESTORE PRESENT UNIT POINTER
3452                                     MOV      (SP)+,R5
3453
3454 017174 012600                        POP     R0                        ;RETRIEVE UNIT NUMBER
3455                                     MOV      (SP)+,R0
3456
3457 017176 010046 017254                  PRINTF  #ABOMSG,R0
3458 017200 012746 000002
3459 017204 012746 000002
3460 017210 010600
3461 017212 104417
3462 017214 062706 000006
3463
3464 017220 104453 003350 017252          GETR5: MOV    #BLKTBL,PTR          ;-->UNIT 0
3465 017220 000016 000016 1$:            MOV    @PTR,R5                ;GET STATUS WORD
3466 017222 012737 000016 1$:            DEC    R0                    ;CORRECT UNIT?
3467 017230 017705 000016 1$:            BMI    2$                    ;YES
3468 017234 005300 000002 017252          ADD    #2,PTR                ;NO,-->NEXT
3469 017236 100404 000002 017252          BR     1$                    ;CONTINUE
3470 017240 062737 000002 017252          BR     1$
3471 017246 000770 000002 017252          2$: RETURN
3472 017250 000207 000002 017252          PTR: .WORD
3473 017252 000000 000002 017252          ABOMSG: .ASCIZ  /%ADROPPED UNIT %D1%N/
3474 017254 045 101 104
3475
3476
3477
3478
3479

```


3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3498
3510
3511
3512
3513
3514

017302
017302

017302
017302
017302 104452

.SBTTL ADD UNIT SECTION

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

BGNAU

LSAU::

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

ENDAU

L10013: TRAP CSAU

```

3573          .SBTTL TEST 1 / DEVICE SELF-DIAGNOSTIC EXECUTION
3574
3575 017304          .NLIST BGNMOD
3576                      ME,BEX
3577
3578 017304          BGNTST
3579 017304          TSTID #TST1
017304 012737 017350 003326
017312 004737 006004
017316 004737 005632
017322 004737 006052
017326 004737 005530
017332 103004
017334 004737 006004
017340 004737 006052
017344
3580 017344          EXIT TST
017344 104432
017346 000136
3581
3582 017350          TST1: TUSELF
017350 012700 027722
017354 112710 000002
017360 112760 000012 000001
017366 112760 000007 000002
017374 105060 000003
017400 005060 000004
017404 005060 000006
017410 005060 000010
017414 005060 000012
017420 012701 000012
017424 005721
017426 012765 000016 000070
017434 004737 013654
017440 010110
017442 012765 000002 000034
017450 012765 000016 000036
017456 012765 000001 000032
017464 004737 006560
017470 032715 000010
017474 001325
3583 017476 005237 003322
3584 017502 000207
3585
3586
3587 017504          INC RETURN DONE
017504          ENDTST
017504 104401
          64$:
          MOV #TST1,TSTTOP ;SAVE ADDR OF TEST
          CALL SETUP ;INIT UNITS TSTPC
          CALL SETDR ;GET 1ST DRVS.
          CALL RUN ;DO TEST
          CALL SWAPDR ;GET NEXT DRVS.
          BCC 64$ ;BR NO 2ND DRVS
          CALL SETUP ;REINIT UNITS TSTPC
          CALL RUN ;REPEAT TEST
          ;DONE
          TRAP C$EXIT
          .WORD L10014-.
          64$:
          MOV #TRBUF,R0 ;FORM COMMAND PACKET
          MOVB #RSCMND,@R0 ;COMMAND FLAG
          MOVB #RSMISZ,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 #RSMISZ,R1 ;GET SIZE
          TST (R1)+ ;+2 FOR CHECKSUM
          MOV #RSSNSZ,SND CNT(R5) ;SIZE TO SEND
          CALL CHKSUM ;FORM CHECKSUM
          MOV R1,(R0) ;INSERT INTO PACKET
          MOV #RSEND,XSFLG(R5) ;EXPECT END.
          MOV #RSNDSZ,XSCNT(R5) ;THIS BIG
          MOV #1,XSPKNT(R5) ;AND 1 PACKET
          ;SEND
          CALL RSVP ;RETURN TO SCHEDULER
          BIT #BIT3,@R5 ;RETRY?(BAD FLAG)
          BNE 64$ ;YES
          L10014:
          TRAP C$ETST

```



```

3590          .SBTTL TEST 2 / SEEK EOT,BOT
3591
3592 017506          BGNTST
3593 017506          TSTID  #TST2
017506          MOV      #TST2,TSTTOP      ;SAVE ADDR OF TEST
017514          CALL     SETUP             ;INIT UNITS TSTPC
017520          CALL     SETDR             ;GET 1ST DRVS.
017524          CALL     RUN              ;DO TEST
017530          CALL     SWAPDR           ;GET NEXT DRVS.
017534          BCC      64$              ;BR NO 2ND DRVS
017536          CALL     SETUP             ;REINIT UNITS TSTPC
017542          CALL     RUN              ;REPEAT TEST
017546          ;DONE
3594 017546          EXIT TST              64$:
017546          TRAP     C$EXIT
017550          .WORD    L10015-.

3595
3596
3597 017552          005004          017742          000064          TST2:  CLR      R4              ;R4=INDEX INTO RECORD TABLE
3598 017554          016465          1$:          MOV      RECDAT(R4),REC(R5) ;GET THE RECORD
3599
3600 017562          TUSEEK  REC(R5),DR(R5) ;SEEK IT

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

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

3601
3602 017720          062704          000002          ADD      #2,R4        ;POINT TO NEXT RECORD
3603 017724          026427          017742          177777          CMP      RECDAT(R4),#-1. ;LAST ONE DONE?
3604 017732          001310          BNE     1$          ;NO-LOOP
3605 017734          005237          00332?          INC      DONE        ;YES-SET DONE FLAG
3606 017740          000207          RETURN
3607

```

MISCELLANEOUS SECTIONS MACRO M1113 25-FEB-82 07:06 PAGE 104-1
TEST 2 / SEEK EOT,BOT

D 8

SEQ 0094

3608	017742	000000	RECDAT: 0.	;BOT
3609	017744	000200	200	;BOT OTHER TRACK
3610	017746	000177	177	;EOT
3611	017750	000377	377	;EOT OTHER TRACK
3612	017752	000400	400	;BOT AGAIN
3613	017754	177777	-1.	
3614	017756		ENDTST	
	017756			
	017756	104401		

L10015: TRAP CSETST

020244 012701 000200
 020250 110160 000001
 020254 010103
 020256 112710 000001
 020262 005720
 020264 116520 000072
 020270 005303
 020272 101374
 020274 012700 027722
 020300 116001 000001
 020304 042701 177400
 020310 010165 000070
 020314 062765 000004 000070
 020322 062701 000002
 020326 004737 013654
 020332 110120
 020334 000301
 020336 110120
 020340 032715 010000
 020344 001412
 020346 012765 000002 000034
 020354 012765 000016 000036
 020362 012765 000001 000032
 020370 000411
 020372 012765 000020 000034
 020400 012765 000001 000036
 020406 012765 000001 000032
 020414 004737 006560
 020420 032715 000010
 020424 001210
 020426 032715 002000
 020432 001004
 020434 162702 000200
 020440 101270
 020442 000502
 020444

020444 012700 027722
 020450 112710 000002
 020454 112760 000012 000001
 020462 112760 000002 000002
 020470 016560 000064 000012
 020476 116560 000060 000004
 020504 105060 000003
 020510 105715
 020512 100002
 020514 105260 000003
 020520 012760 001000 000010
 020526 112760 000020 000005
 020534 005060 000006
 020540 012701 000012
 020544 005721
 020546 012765 000016 000070
 020554 004737 013654

65\$: MOV #128.,R1 ;USE 128. BYTES
 66\$: MOV R1,1(R0) ;COPY COUNT TO BUFFER
 MOV R1,R3 ;R3=COUNTER TO LOAD BUFFER
 MOVB #RSDATA, @R0 ;FLAG FIRST
 TST (R0)+ ;SKIP COUNT
 67\$: MOVB PATTEN(R5), (R0)+ ;INSERT DATA
 DEC R3 ;MORE?
 BHI 67\$;YES
 MOV #TRBUF, R0 ;-->TOP AGAIN
 MOVB 1(R0), R1 ;GET COUNT
 BIC #177400, R1 ;ZERO SIGN EXTEND
 MOV R1, SDCNT(R5) ;HOW MANY TO SEND PLUS
 ADD #4, SDCNT(R5) ;FLAG, COUNT, CHKSUM
 ADD #2, R1 ;COMPENSATE FOR FLAG + COUNT
 CALL CHKSUM ;FOR CHECKSUM CALC.
 MOV R1, (R0)+ ;CHKSUM INTO PACKET
 SWAB R1 ;EVEN ON AN ODD
 MOV R1, (R0)+ ;BYTE BOUNDARY
 BIT #BIT12, @R5 ;LAST DATA PACKET?
 BEQ 68\$;NO
 MOV #RSEND, XSFLG(R5) ;YES-EXPECT 'END'
 MOV #RSNDSZ, XSCNT(R5) ;OF THIS SIZE
 MOV #1, XSPKNT(R5) ;AND 1 PACKET
 BR 69\$;SEND
 68\$: MOV #RSCONT, XSFLG(R5) ;(NOT LAST), EXPECT 'CONTINUE'
 MOV #1, XSCNT(R5) ;AND 1 BYTE
 MOV #1, XSPKNT(R5) ;AND 1 PACKET
 69\$: CALL RSVB ;SEND PACKET
 ;AND RETURN TO SCHEDULER
 BIT #BIT3, @R5 ;FLAG BYTE RETRY?
 BNE 72\$;YES
 BIT #BIT10, @R5 ;RETRY DATA ERROR?
 BNE 70\$;YES
 SUB #128., R2 ;NO, MORE DATA TO SEND?
 BHI 64\$;YES
 BR 71\$;NO
 70\$: TURTRY REC(R5), #512., @R(R5) ;RETRY HERE
 76\$: MOV #TRBUF, R0 ;FORM CMD PACK:
 MOVB #RSCMD, @R0 ;MESSAGE PACK TYPE
 MOVB #RSMISZ, 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 77\$;NO
 INCB 3(R0) ;YES-CHANGE THRESHOLD
 77\$: MOV #512., 8.(R0) ;# BYTES DESIRED
 MOVB #020, 5.(R0) ;MAINTENANCE MODE
 CLR 6.(R0) ;NO SEQUENCE #
 MOV #RSMISZ, R1 ;SIZE OF PACKET
 TST (R1)+ ;PLUS FLAG+COUNT INTO R1
 MOV #RSSNSZ, SDCNT(R5) ;SET UP SIZE TO SEND
 CALL CHKSUM ;FORM CHECKSUM R1=COUNT

020560 010110
 020562 012701 001000
 020566 012703 000034
 020572 060503
 020574 005002
 020576 005202
 020600 012723 000001
 020604 012723 000204
 020610 162701 000200
 020614 101401
 020616 000767
 020620 005202
 020622 010265 000032
 020626 012723 000002
 020632 012713 000016
 020636 004737 006560

```

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
  
```

3632 020652

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

020652 012700 027722
 020656 112710 000002
 020662 112760 000012 000001
 020670 112760 000002 000002
 020676 016560 000064 000012
 020704 116560 000060 000004
 020712 112760 000000 000003
 020720 012760 001000 000010
 020726 112760 000020 000005
 020734 005060 000006
 020740 012701 000012
 020744 005721
 020746 012765 000016 000070
 020754 004737 013654
 020760 010110

```

82$: MOV #TRBUF,R0 ;FORM CMD PACK:
MOV #RSCMD,R0 ;MESSAGE PACK TYPE
MOV #RSMISZ,1(R0) ;THIS BIG
MOV #RSSRD,2(R0) ;OP CODE IS READ
MOV REC(R5),10.(R0) ;THIS RECORD
MOV DR(R5),4.(R0) ;THIS DRIVE
MOV #0,3.(R0) ;VERIFY
MOV #512.,8.(R0) ;TOTAL BYTES TO READ
MOV #020,5.(R0) ;MAINTENANCE MODE
CLR 6.(R0) ;NO SEQUENCE #
MOV #RSMISZ,R1 ;GET SIZE OF PACKET
TST (R1)+ ;+2 FOR CHECKSUM
MOV #RSSNSZ,SNDCNT(R5) ;SIZE TO SEND
CALL CHKSUM ;FORM CHECKSUM R1=COUNT
MOV R1,(R0) ;INSERT CHECKSUM

MOV #512.,R1 ;SET EXPECTATIONS:
;CALC # OF DATA PACKETS TO EXPECT:
MOV #XSFLG,R3 ;GET OFFSET
ADD R5,R3 ;ABS. ADDR. OF XSFLG
CLR R2 ;PRESET AS NONE
78$: INC R2 ;# PACKETS EXPECTED
MOV #RSDATA,(R3)+ ;LOAD XSFLG
MOV #132.,(R3)+ ;AND EXPECTED COUNT
SUB #128.,R1 ;NEG RESULT LAST TIME
BLOS 80$ ;LAST TIME
BR 78$ ;MORE TO DO
80$: INC R2 ;ADD ONE FOR END PACK
MOV R2,XSPKMM(R5) ;SAVE # PACKETS TO EXPECT
MOV #RSEND,(R3)+ ;EXPECT AN END ALSO...
MOV #RSNDSZ,(R3) ;THIS BIG-14. BYTES
CALL RSVP ;SEND
;AND RETURN TO SCHEDULER
  
```

020762 012701 001000
 020766 012703 000034
 020772 060503
 020774 005002
 020776 005202
 021000 012723 000001
 021004 012723 000204
 021010 162701 000200
 021014 101401
 021016 000767
 021020 005202
 021022 010265 000032
 021026 012723 000002
 021032 012713 000016
 021036 004737 006560

021042	032715	002010		81\$:	BIT	#BIT10!BIT3,@R5	:RETRY?
021046	001500				BEQ	79\$	:NO.
021050					TURTRY	REC(R5),#512.,DR(R5)	:YES
021050	012700	027722		86\$:	MOV	#TRBUF,R0	:FORM CMND PACK:
021054	112710	000002			MOVB	#RSCMND,@R0	:MESSAGE PACK TYPE
021060	112760	000012	000001		MOVB	#RSMISZ,1(R0)	:THIS BIG
021066	112760	000002	000002		MOVB	#RSSRD,2(R0)	:OP CODE-READ
021074	016560	000064	000012		MOV	REC(R5),10.(R0)	:THIS RECORD
021102	116560	000060	000004		MOVB	DR(R5),4.(R0)	:THIS DRIVE
021110	105060	000003			CLRB	3(R0)	:PRESET NORM THRESHOLD
021114	105715				TSTB	@R5	:REDUCED?
021116	100002				BPL	87\$	:NO
021120	105260	000003			INCB	3(R0)	:YES-CHANGE THRESHOLD
021124	012760	001000	000010	87\$:	MOV	#512.,8.(R0)	:# BYTES DESIRED
021132	112760	000020	000005		MOVB	#020,5.(R0)	:MAINTENANCE MODE
021140	005060	000006			CLR	6.(R0)	:NO SEQUENCE #
021144	012701	000012			MOV	#RSMISZ,R1	:SIZE OF PACKET
021150	005721				TST	(R1)+	:PLUS FLAG+COUNT INTO R1
021152	012765	000016	000070		MOV	#RSSNSZ,SND CNT(R5)	:SET UP SIZE TO SEND
021160	004737	013654			CALL	CHKSUM	:FORM CHECKSUM R1=COUNT
021164	010110				MOV	R1,(R0)	:INSERT IN PACKET
021166	012701	001000			MOV	#512.,R1	:SET EXPECTATIONS:
021172	012703	000034					:CALC # OF DATA PACKETS TO EXPECT
021176	060503				MOV	#XSFLG,R3	:OFFSET OF FLAG
021200	005002				ADD	R5,R3	:ABS. ADDR. OF XSFLG
021202	005202				CLR	R2	:PRESET
021204	012723	000001		83\$:	INC	R2	:# PACKETS EXPECTED
021210	012723	000204			MOV	#RSDATA,(R3)+	:LOAD XSFLG
021214	162701	000200			MOV	#132.,(R3)+	:AND EXPECT COUNT
021220	101401				SUB	#128.,R1	:NEG RESULT LAST TIME
021222	000767				BLOS	85\$	:LAST TIME!
021224	005202				BR	83\$	:MORE TO DO
021226	010265	000032		85\$:	INC	R2	:ADD ONE FOR END PACK
021232	012723	000002			MOV	R2,XSPKMM(R5)	:SAVE # PACKETS TO EXPECT
021236	012713	000016			MOV	#RSEND,(R3)+	:EXPECT AN END
					MOV	#RSNDSZ,(R3)	:THIS BIG-14. BYTES
021242	004737	006560			CALL	RSVP	:SEND
							:AND RETURN TO SCHEDULER
3633	021252	062704	000002		ADD	#2,R4	:POINT TO NEXT DATA
3634	021256	005764	022742		TST	TST3PT(R4)	:END?
3635	021262	001402			BEQ	2\$	:YES
3636	021264	000137	020040		JMP	1\$	:NO-WRITE, READ NEW DATA
3637	021270	005004		2\$:	CLR	R4	:POINT TO FIRST DATA
3638	021272	062765	000200	000064	ADD	#200,REC(R5)	:BUT NOW USE ADJACENT RECORD
3639	021300	032765	001000	000064	BIT	#1000,REC(R5)	:ALL ADJACENT RECORDS DONE?
3640	021306	001002			BNE	3\$	:YES
3641	021310	000137	020040		JMP	1\$	:NO-WRITE, READ AT NEW RECORD
3642	021314	162765	001000	000064	SUB	#1000,REC(R5)	:RESTORE TO NEXT RECORD
3643	021322	066565	000066	000064	ADD	TMP(R5),REC(R5)	:HALF INTO REST OF TAPE
3644	021330	006265	000066		ASR	TMP(R5)	:HALF OF HALF FOR NEXT TIME
3645	021334	103402			BCS	4\$	:DONE?

MISCELLANEOUS SECTIONS MACRO M1113 25-FEB-82 07:06 PAGE 106-4
TEST 3 / HIGH ACTIVITY WRITE/READ (512 BYTE/BLOCK MODE)

I 8

SEQ 0099

3646 021336 000137 020040
3647 021342 005237 003322
3648 021346 000207
3649 021350
021350
021350 104401
3650

4\$: JMP 1\$
INC DONE
RETURN
ENDTST

:NO
:YES-SET FLAG

L10016: TRAP C\$ETST

```

3652          .SBTTL TEST 4 / HIGH ACTIVITY WRITE/READ (128 BYTE/BLOCK MODE)
3653
3654          : WRITE THEN READ VARYING DATA FOR ALL PHYSICALLY ADJACENT BLOCKS AROUND
3655          : A RECORD, GO HALF-WAY INTO REMAINING TAPE REPEAT UNTIL EOT.
3656
3657          BGNTST
3658          TSTID #TST4
3659          T4::
3660          MOV #TST4,TSTTOP ;SAVE ADDR OF TEST
3661          CALL SETUP ;INIT UNITS TSTPC
3662          CALL SETDR ;GET 1ST DRVS.
3663          CALL RUN ;DO TEST
3664          CALL SWAPDR ;GET NEXT DRVS.
3665          BCC 64$ ;BR NO 2ND DRVS
3666          CALL SETUP ;REINIT UNITS TSTPC
3667          CALL RUN ;REPEAT TEST
3668          ;DONE
3669          EXIT TST
3670          64$:
3671          TRAP C$EXIT
3672          .WORD L10017-.
3673
3674          TST4: MOV #400,TMP(R5) ;INIT TO HALF OF REMAINING
3675          CLR R4 ;FOR INDEX INTO DATA TABLE
3676          CLR REC(R5) ;START AT RECORD 0
3677          1$: MOV TST3PT(R4),PATTEN(R5) ;GET DATA
3678          TUWRIT PATTEN(R5),REC(R5),#128,DR(R5),#BIT7
3679          72$: MOV #TRBUF,R0 ;MAKE COMMAND PACKET:
3680          MOV #RSCMND,R0 ;COMMAND FLAG
3681          MOV #RSMISZ,1(R0) ;THIS SIZE
3682          MOV #RSSWR,2(R0) ;INSERT OP CODE-WRITE
3683          MOV #BIT7,3(R0) ;VERIFY (1 OR 0)
3684          MOV DR(R5),4(R0) ;DRIVE #
3685          MOV #020,5(R0) ;MAINTENANCE MODE SWITCH
3686          CLR 6(R0) ;NO SEQUENCE #
3687          MOV #128,8(R0) ;TOTAL COUNT TO WRITE
3688          MOV REC(R5),10(R0) ;AT RECORD N
3689          MOV #RSMISZ,R1 ;THE PACKET SIZE PLUS+2
3690          TST (R1)+ ;(FLAG AND COUNT) INTO R1
3691          MOV #RSSNSZ,SND CNT(R5) ;LOAD THE SIZE TO SEND
3692          CALL CHKSUM ;R0 --> R1=COUNT
3693          MOV R1,(R0) ;PUT CHKSUM IN PACKET
3694          ;SET UP EXPECTATIONS:
3695          MOV #RSCONT,XSFLG(R5) ;THE FLAG
3696          MOV #1,XSCNT(R5) ;THE COUNT
3697          MOV #1,XSPKNM(R5) ;THE # PACKETS EXPECTED
3698          MOV #128,R2 ;GET # OF DATA BYTES
3699          CALL RSVP ;SEND (AND RETURN TO SCHEDULER)
3700          BIT #BIT3,R5 ;FLAG BYTE ERROR?
3701          BNE 72$ ;YES
3702          BIC #BIT12,R5 ;FLAG FOR LAST PACKET
3703          MOV #TRBUF,R0 ;POINT TO TOP OF BUFFER AGAIN
3704          CMP R2,#128. ;START DATA PACKET(S)
3705          BHI 65$ ;#128. > 128.!
3706          MOV R2,R1 ;#128.<128.
3707          BIS #BIT12,R5 ;SO LAST PACKET NOW
3708          BR 66$ ;USE REMAINING COUNT
3709
3710          64$:
3711          MOV #RSCONT,XSFLG(R5) ;THE FLAG
3712          MOV #1,XSCNT(R5) ;THE COUNT
3713          MOV #1,XSPKNM(R5) ;THE # PACKETS EXPECTED
3714          MOV #128,R2 ;GET # OF DATA BYTES
3715          CALL RSVP ;SEND (AND RETURN TO SCHEDULER)
3716          BIT #BIT3,R5 ;FLAG BYTE ERROR?
3717          BNE 72$ ;YES
3718          BIC #BIT12,R5 ;FLAG FOR LAST PACKET
3719          MOV #TRBUF,R0 ;POINT TO TOP OF BUFFER AGAIN
3720          CMP R2,#128. ;START DATA PACKET(S)
3721          BHI 65$ ;#128. > 128.!
3722          MOV R2,R1 ;#128.<128.
3723          BIS #BIT12,R5 ;SO LAST PACKET NOW
3724          BR 66$ ;USE REMAINING COUNT

```



```

021636 012701 000200
021642 110160 000001
021646 010103
021650 112710 000001
021654 005720
021656 116520 000072
021662 005303
021664 101374
021666 012700 027722
021672 116001 000001
021676 042701 177400
021702 010165 000070
021706 062765 000004 000070
021714 062701 000002
021720 004737 013654
021724 110120
021726 000301
021730 110120
021732 032715 010000
021736 001412
021740 012765 000002 000034
021746 012765 000016 000036
021754 012765 000001 000032
021762 000411
021764 012765 000020 000034
021772 012765 000001 000036
022000 012765 000001 000032
022006 004737 006560

022012 032715 000010
022016 001210
022020 032715 002000
022024 001004
022026 162702 000200
022032 101270
022034 000502
022036

022036 012700 027722
022042 112710 000002
022046 112760 000012 000001
022054 112760 000002 000002
022062 016560 000064 000012
022070 116560 000060 000004
022076 105060 000003
022102 105715
022104 100002
022106 105260 000003
022112 012760 000200 000010
022120 112760 000020 000005
022126 005060 000006
022132 012701 000012
022136 005721
022140 012765 000016 000070

022146 004737 013654
  
```

```

65$: MOV #128.,R1 ;USE 128. BYTES
66$: MOVB R1,1(R0) ;COPY COUNT TO BUFFER
      MOV R1,R3 ;R3=COUNTER TO LOAD BUFFER
      MOVB #RSDATA,@R0 ;FLAG FIRST
      TST (R0)+ ;SKIP COUNT
67$: MOVB PATTEN(R5),(R0)+ ;INSERT DATA
      DEC R3 ;MORE?
      BHI 67$ ;YES
      MOV #TRBUF,R0 ;-->TOP AGAIN
      MOVB 1(R0),R1 ;GET COUNT
      BIC #177400,R1 ;ZERO SIGN EXTEND
      MOV R1,SND CNT(R5) ;HOW MANY TO SEND PLUS
      ADD #4,SND CNT(R5) ;FLAG,COUNT,CHKSUM
      ADD #2,R1 ;COMPENSATE FOR FLAG + COUNT
      CALL CHKSUM ;FOR CHECKSUM CALC.
      MOVB R1,(R0)+ ;CHKSUM INTO PACKET
      SWAB R1 ;EVEN ON AN ODD
      MOVB R1,(R0)+ ;BYTE BOUNDARY
      BIT #BIT12,@R5 ;LAST DATA PACKET?
      BEQ 68$ ;NO
      MOV #RSEND,XSFLG(R5) ;YES-EXPECT 'END'
      MOV #RSNDSZ,XSCNT(R5) ;OF THIS SIZE
      MOV #1,XSPKNT(R5) ;AND 1 PACKET
      BR 69$ ;SEND
68$: MOV #RSCONT,XSFLG(R5) ;(NOT LAST), EXPECT 'CONTINUE'
      MOV #1,XSCNT(R5) ;AND 1 BYTE
      MOV #1,XSPKNT(R5) ;AND 1 PACKET
69$: CALL R5VP ;SEND PACKET
      ;AND RETURN TO SCHEDULER
      BIT #BIT3,@R5 ;FLAG BYTE RETRY?
      BNE 72$ ;YES
      BIT #BIT10,@R5 ;RETRY DATA ERROR?
      BNE 70$ ;YES
      SUB #128.,R2 ;NO, MORE DATA TO SEND?
      BHI 64$ ;YES
      BR 71$ ;NO
70$: TURTRY REC(R5),#128.,DR(R5) ;RETRY HERE

76$: MOV #TRBUF,R0 ;FORM CMD PACK:
      MOVB #RSCMD,@R0 ;MESSAGE PACK TYPE
      MOVB #RSMISZ,1(R0) ;THIS BIG
      MOVB #RSSRD,2(R0) ;OP CODE-READ
      MOV REC(R5),10.(R0) ;THIS RECORD
      MOVB DR(R5),4.(R0) ;THIS DRIVE
      CLRB 3(R0) ;PRESET NORM THRESHOLD
      TSTB @R5 ;REDUCED?
      BPL 77$ ;NO
      INCB 3(R0) ;YES-CHANGE THRESHOLD
77$: MOV #128.,8.(R0) ;# BYTES DESIRED
      MOVB #020,5.(R0) ;MAINTENANCE MODE
      CLR 6.(R0) ;NO SEQUENCE #
      MOV #RSMISZ,R1 ;SIZE OF PACKET
      TST (R1)+ ;PLUS FLAG+COUNT INTO R1
      MOV #RSSNSZ,SND CNT(R5) ;SET UP SIZE TO SEND

      CALL CHKSUM ;FORM CHECKSUM R1=COUNT
  
```

022152 010110
 022154 012701 000200
 022160 012703 000034
 022164 060503
 022166 005002
 022170 005202
 022172 012723 000001
 022176 012723 000204
 022202 162701 000200
 022206 101401
 022210 000767
 022212 005202
 022214 010265 000032
 022220 012723 000002
 022224 012713 000016
 022230 004737 006560

```

MOV      R1,(R0)          ;INSERT IN PACKET
MOV      #128.,R1         ;SET EXPECTATIONS:
                          ;CALC # OF DATA PACKETS TO EXPECT
MOV      #XSFLG,R3        ;GET OFFSET
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,XSPKMN(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
  
```

3667 022244

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

022244 012700 027722
 022250 112710 000002
 022254 112760 000012 000001
 022262 112760 000002 000002
 022270 016560 000064 000012
 022276 116560 000060 000004
 022304 112760 000200 000003
 022312 012760 000200 000010
 022320 112760 000020 000005
 022326 005060 000006
 022332 012701 000012
 022336 005721
 022340 012765 000016 000070
 022346 004737 013654
 022352 010110
 022354 012701 000200
 022360 012703 000034
 022364 060503
 022366 005002
 022370 005202
 022372 012723 000001
 022376 012723 000204
 022402 162701 000200
 022406 101401
 022410 000767
 022412 005202
 022414 010265 000032
 022420 012723 000002
 022424 012713 000016
 022430 004737 006560

```

82$:     MOV      #TRBUF,R0 ;FORM CMND PACK:
          MOVB     #RSCMD,@R0 ;MESSAGE PACK TYPE
          MOVB     #RSMISZ,1(R0) ;THIS BIG
          MOVB     #RSSRD,2(R0) ;OP CODE IS READ
          MOV      REC(R5),10.(R0) ;THIS RECORD
          MOVB     DR(R5),4.(R0) ;THIS DRIVE
          MOVB     #BIT7,3.(R0) ;VERIFY
          MOV      #128.,8.(R0) ;TOTAL BYTES TO READ
          MOVB     #020,5.(R0) ;MAINTENANCE MODE
          CLR      6.(R0)      ;NO SEQUENCE #
          MOV      #RSMISZ,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      #128.,R1   ;SET EXPECTATIONS:
                          ;CALC # OF DATA PACKETS TO EXPECT:
          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     80$        ;LAST TIME
          BR       78$        ;MORE TO DO
          INC      R2        ;ADD ONE FOR END PACK
          MOV      R2,XSPKMN(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
          78$:
          80$:
  
```

022434	032715	002010		81\$:	BIT	#BIT10!BIT3,@R5	:RETRY?
022440	001500				REQ	79\$	:NO.
022442					URTRY	REC(R5),#128.,DR(R5)	:YES
022442	012700	027722		86\$:	MOV	#TRBUF,R0	:FORM CMND PACK:
022446	112710	000002			MOVB	#RSCMND,@R0	:MESSAGE PACK TYPE
022452	112760	000012	000001		MOVB	#RSMISZ,1(R0)	:THIS BIG
022460	112760	000002	000002		MOVB	#RSSRD,2(R0)	:OP CODE-READ
022466	016560	000064	000012		MOV	REC(R5),10.(R0)	:THIS RECORD
022474	116560	000060	000004		MOVB	DR(R5),4.(R0)	:THIS DRIVE
022502	105060	000003			CLRB	3(R0)	:PRESET NORM THRESHOLD
022506	105715				TSTB	@R5	:REDUCED?
022510	100002				BPL	87\$	:NO
022512	105260	000003			INCB	3(R0)	:YES-CHANGE THRESHOLD
022516	012760	000200	000010	87\$:	MOV	#128.,8.(R0)	:# BYTES DESIRED
022524	112760	000020	000005		MOVB	#020,5.(R0)	:MAINTENANCE MODE
022532	005060	000006			CLR	6.(R0)	:NO SEQUENCE #
022536	012701	000012			MOV	#RSMISZ,R1	:SIZE OF PACKET
022542	005721				TST	(R1)+	:PLUS FLAG+COUNT INTO R1
022544	012765	000016	000070		MOV	#RSSNSZ,SND CNT(R5)	:SET UP SIZE TO SEND
022552	004737	013654			CALL	CHKSUM	:FORM CHECKSUM R1=COUNT
022556	010110				MOV	R1,(R0)	:INSERT IN PACKET
022560	012701	000200			MOV	#128.,R1	:SET EXPECTATIONS:
022564	012703	000034			MOV	#XSFLG,R3	:CALC # OF DATA PACKETS TO EXPECT
022570	060503				ADD	R5,R3	:OFFSET OF FLAG
022572	005002				CLR	R2	:ABS. ADDR. OF XSFLG
022574	005202			83\$:	INC	R2	:PRESET
022576	012723	000001			MOV	#RSDATA,(R3)+	:# PACKETS EXPECTED
022602	012723	000204			MOV	#132.,(R3)+	:LOAD XSFLG
022606	162701	000200			SUB	#128.,R1	:AND EXPECT COUNT
022612	101401				BLOS	85\$	:NEG RESULT LAST TIME
022614	000767				BR	83\$	:LAST TIME!
022616	005202			85\$:	INC	R2	:MORE TO DO
022620	010265	000032			MOV	R2,XSPKNM(R5)	:ADD ONE FOR END PACK
022624	012723	000002			MOV	#RSEND,(R3)+	:SAVE # PACKETS TO EXPECT
022630	012713	000016			MOV	#RSNDSZ,(R3)	:EXPECT AN END
022634	004737	006560			CALL	RSVP	:THIS BIG-14. BYTES
							:SEND
							:AND RETURN TO SCHEDULER
3668	022644	062704	000002		ADD	#2,R4	:POINT TO NEXT DATA
3669	022650	005764	022742		TST	TST3PT(R4)	:END?
3670	022654	001402			BEQ	2\$	:YES
3671	022656	000137	021432		JMP	1\$	:NO-WRITE, READ NEW DATA
3672	022662	005004		2\$:	CLR	R4	:POINT TO FIRST DATA
3673	022664	062765	001000	000064	ADD	#1000,REC(R5)	:BUT NOW USE ADJACENT RECORD
3674	022672	032765	004000	000064	BIT	#4000,REC(R5)	:ALL ADJACENT RECORDS DONE?
3675	022700	001002			BNE	3\$	:YES
3676	022702	000137	021432		JMP	1\$	:NO-WRITE, READ AT NEW RECORD
3677	022706	162765	004000	000064	SUB	#4000,REC(R5)	:RESTORE TO NEXT RECORD
3678	022714	066565	000066	000064	ADD	TMP(R5),REC(R5)	:HALF INTO REST OF TAPE
3679	022722	006265	000066		ASR	TMP(R5)	:HALF OF HALF FOR NEXT TIME
3680	022726	103402			BCS	4\$	:DONE?

MISCELLANEOUS SECTIONS MACRO M1113 25-FEB-82 07:06 PAGE 107-4
TEST 4 / HIGH ACTIVITY WRITE/READ (128 BYTE/BLOCK MODE)

SEQ 0104

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

L10017: TRAP C\$ETST

```

3694
3695
3696
3697 022756          .SBTTL TEST 5 / WRITE SELECTED NUMBER OF BLOCKS
                                BGNTST
                                (STI) #TST5
                                MOV #TST5,TSTTOP
                                CALL SETUP
                                CALL SETDR
                                CALL RUN
                                CALL SWAPDR
                                BCC 64$
                                CALL SETUP
                                CALL RUN
                                T5::
                                ;SAVE ADDR OF TEST
                                ;INIT UNITS TSTPC
                                ;GET 1ST DRVS.
                                ;DO TEST
                                ;GET NEXT DRVS.
                                ;BR NO 2ND DRVS
                                ;REINIT UNITS TSTPC
                                ;REPEAT TEST
                                ;DONE
3698 022756 012737 023022 003326
022756 004737 006004
022764 004737 005632
022770 004737 006052
022774 004737 006052
023000 004737 005530
023004 103004
023006 004737 006004
023012 004.37 006052
023016
3699 023016          EXIT TST
023016 104432
023020 000724
                                TRAP C$EXIT
                                .WORD L10020-.
3700
3701
3702 023022 005065 000064 000066 TST5: CLR REC(R5) ;START AT REC 0
3703 023026 013765 003310 000066 MOV TAPLEN,TMP(R5) ;GET THE # OF BLOCKS PER TRACK
3704 023034 005065 000062 000072 1$: CLR TRK(R5) ;TRK(R5)=1ST OR 2ND PASS COUNTER
3705 023040 016565 000064 000072 1$: MOV REC(R5),PATTEN(R5) ;USE RECORD NO. FOR DATA
3706 023046 005737 002220 TST DRVCHK ;ADD DR #?
3707 023052 001403 BEQ 10$ ;NO
3708 023054 066565 000060 000072 10$: ADD DR(R5),PATTEN(R5) ;YES, ADD DRIVE ID
3709 023062 012700 027722 10$: TUWRIT PATTEN(R5),REC(R5),#512.,DR(R5),#0
                                72$: MOV #TRBUF,R0 ;MAKE COMMAND PACKET:
                                MOV #RSCMND,@R0 ;COMMAND FLAG
                                MOV #RSMISZ,1(R0) ;THIS SIZE
                                MOV #RSSWR,2(R0) ;INSERT OP CODE-WRITE
                                MOV #0,3.(R0) ;VERIFY (1 OR 0)
                                MOV DR(R5),4.(R0) ;DRIVE #
                                MOV #020,5.(R0) ;MAINTENANCE MODE SWITCH
                                CLR 6.(R0) ;NO SEQUENCE #
                                MOV #512,8.(R0) ;TOTAL COUNT TO WRITE
                                MOV REC(R5),10.(R0) ;AT RECORD N
                                MOV #RSMISZ,R1 ;THE PACKET SIZE PLUS+2
                                TST (R1)+ ;(FLAG AND COUNT) INTO R1
                                MOV #RSSNSZ,SNDcnt(R5) ;LOAD THE SIZE TO SEND
                                CALL CHKSUM ;R0 --> R1=COUNT
                                MOV R1,(R0) ;PUT CHKSUM IN PACKET
                                ;SET UP EXPECTATIONS:
                                MOV #RSCONT,XSFLG(R5) ;THE FLAG
                                MOV #1,XSCNT(R5) ;THE COUNT
                                MOV #1,XSPKMM(R5) ;THE # PACKETS EXPECTED
                                MOV #512.,R2 ;GET # OF DATA BYTES
                                CALL RSVP ;SEND (AND RETURN TO SCHEDULER)
                                BIT #BIT3,@R5 ;FLAG BYTE ERROR?
                                BNE 72$ ;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$:
023062 012700 027722
023066 112710 000002
023072 112760 000012 000001
023100 112760 000003 000002
023106 112760 000000 000003
023114 116560 000060 000004
023122 112760 000020 000005
023130 005060 000006
023134 012760 001000 000010
023142 016560 000064 000012
023150 012701 000012
023154 005721
023156 012765 000016 000070
023164 004737 013654
023170 010110
023172 012765 000020 000034
023200 012765 000001 000036
023206 012765 000001 000032
023214 012702 001000
023220 004737 006560
023224 032715 000010
023230 001314
023232 042715 010000
023236 012700 027722
023242 020227 000200
023246 101004
023250 010201
023252 052715 010000

```

023256	000402		
023260	012701	000200	
023264	110160	000001	
023270	010103		
023272	112710	000001	
023276	005720		
023300	116520	000072	
023304	005303		
023306	101374		
023310	012700	027722	
023314	116001	000001	
023320	042701	177400	
023324	010165	000070	
023330	062765	000004	000070
023336	062701	000002	
023342	004737	013654	
023346	110120		
023350	000301		
023352	110120		
023354	032715	010000	
023360	001412		
023362	012765	000002	000034
023370	012765	000016	000036
023376	012765	000001	000032
023404	000411		
023406	012765	000020	000034
023414	012765	000001	000036
023422	012765	000001	000032
023430	004737	006560	
023434	032715	000010	
023440	001210		
023442	032715	002000	
023446	001004		
023450	162702	000200	
023454	101270		
023456	000502		
023460			

023460	012700	027722	
023464	112710	000002	
023470	112760	000012	000001
023476	112760	000002	000002
023504	016560	000064	000012
023512	116560	000060	000004
023520	105060	000003	
023524	105715		
023526	100002		
023530	105260	000003	
023534	012760	001000	000010
023542	112760	000020	000005
023550	005060	000006	
023554	012701	000012	
023560	005721		
023562	012765	000016	000070

	BR	66\$	:USE REMAINING COUNT
65\$:	MOV	#128.,R1	:USE 128. BYTES
66\$:	MOVB	R1,1(R0)	:COPY COUNT TO BUFFER
	MOV	R1,R3	:R3=COUNTER TO LOAD BUFFER
	MOVB	#RSDATA,@R0	:FLAG FIRST
	TST	(R0)+	:SKIP COUNT
67\$:	MOVB	PATTEN(R5),(R0)+	:INSERT DATA
	DEC	R3	:MORE?
	BHI	67\$	:YES
	MOV	#TRBUF,R0	:-->TOP AGAIN
	MOVB	1(R0),R1	:GET COUNT
	BIC	#177400,R1	:ZERO SIGN EXTEND
	MOV	R1,SNDCNT(R5)	:HOW MANY TO SEND PLUS
	ADD	#4,SNDCNT(R5)	:FLAG,COUNT,CHKSUM
	ADD	#2,R1	:COMPENSATE FOR FLAG + COUNT
	CALL	CHKSUM	:FOR CHECKSUM CALC.
	MOVB	R1,(R0)+	:CHKSUM INTO PACKET
	SWAB	R1	:EVEN ON AN ODD
	MOVB	R1,(R0)+	:BYTE BOUNDARY
	BIT	#BIT12,@R5	:LAST DATA PACKET?
	BEQ	68\$	:NO
	MOV	#RSEND,XSFLG(R5)	:YES-EXPECT 'END'
	MOV	#RSNDSZ,XSCNT(R5)	:OF THIS SIZE
	MOV	#1,XSPKMM(R5)	:AND 1 PACKET
	BR	69\$	:SEND
68\$:	MOV	#RSCONT,XSFLG(R5)	:(NOT LAST), EXPECT 'CONTINUE'
	MOV	#1,XSCNT(R5)	:AND 1 BYTE
	MOV	#1,XSPKMM(R5)	:AND 1 PACKET
69\$:	CALL	RSVP	:SEND PACKET
			:AND RETURN TO SCHEDULER
	BIT	#BIT3,@R5	:FLAG BYTE RETRY?
	BNE	72\$	:YES
	BIT	#BIT10,@R5	:RETRY DATA ERROR?
	BNE	70\$	:YES
	SUB	#128.,R2	:NO, MORE DATA TO SEND?
	BHI	64\$	:YES
	BR	71\$	:NO
70\$:	TURTRY	REC(R5),#512.,DR(R5)	:RETRY HERE
76\$:	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	77\$	:NO
	INCB	3(R0)	:YES-CHANGE THRESHOLD
77\$:	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,SNDCNT(R5)	:SET UP SIZE TO SEND

023570 004737 013654
 023574 010110
 023576 012701 001000
 023602 012703 000034
 023606 060503
 023610 005002
 023612 005202
 023614 012723 000001
 023620 012723 000204
 023624 162701 000200
 023630 101401
 023632 000767
 023634 005202
 023636 010265 000032
 023642 012723 000002
 023646 012713 000016
 023652 004737 006560

73\$:

75\$:

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

3710 023666 005365 000066
 3711 023672 001404
 3712 023674 005265 000064
 3713 023700 000137 023040
 3714 023704 005765 000062
 3715 023710 001012
 3716 023712 005265 000062
 3717 023716 013765 003334 000064
 3718 023724 013765 003310 000066
 3719 023732 000137 023040
 3720 023736 005237 003322
 3721 023742 000207
 3722
 3723 023744
 023744
 023744 104401

2\$:

TST5EX:

DEC TMP(R5)
 BEQ 2\$
 INC REC(R5)
 JMP 1\$
 TST TRK(R5)
 BNE TST5EX
 INC TRK(R5)
 MOV SECRC,REC(R5)
 MOV TAPLEN,TMP(R5)
 JMP 1\$
 INC DONE
 RETURN
 ENDTST

;DO ALL RECORDS FOR THIS TRACK?
 ;YES-GET OTHER TRACK
 ;NO-ONTO NEXT RECORD
 ;EXECUTE THE WRITE
 ;DONE 2 TRACKS?
 ;YES-EXIT
 ;NO-SET FLAG FOR NEXT PASS
 ;GET NEW STARTING BLOCK #
 ;RESET # OF BLOCKS
 ;AND EXECUTE
 ;DONE
 ;RETURN

L10020: TRAP CSETST

024222	010265	000032			MOV	R2,XSPKMM(R5)	:SAVE # PACKETS TO EXPECT
024226	012723	000002			MOV	#RSEND,(R3)+	:EXPECT AN END ALSO...
024232	012713	000016			MOV	#RSNDSZ,(R3)	:THIS BIG-14. BYTES
024236	004737	006560			CALL	RSVP	:SEND
							:AND RETURN TO SCHEDULER
024242	032715	002010		67\$:	BIT	#BIT10!BIT3,@R5	:RETRY?
024246	001500				BEQ	65\$	:NO
024250					TURTRY	REC(R5),#512.,DR(R5)	:YES
				72\$:	MOV	#TRBUF,R0	:FORM CMND PACK:
024250	012700	027722			MOVB	#RSCMND,@R0	:MESSAGE PACK TYPE
024254	112710	000002			MOVB	#RSMSIZ,1(R0)	:THIS BIG
024260	112760	000012	000001		MOVB	#RSSRD,2(R0)	:OP CODE-READ
024266	112760	000002	000002		MOV	REC(R5),10.(R0)	:THIS RECORD
024274	016560	000064	000012		MOVB	DR(R5),4.(R0)	:THIS DRIVE
024302	116560	000060	000004		CLRB	3(R0)	:PRESET NORM THRESHOLD
024310	105060	000003			TSTB	@R5	:REDUCED?
024314	105715				BPL	73\$	:NO
024316	100002				INCB	3(R0)	:YES-CHANGE THRESHOLD
024320	105260	000003		73\$:	MOV	#512.,8.(R0)	:# BYTES DESIRED
024324	012760	001000	000010		MOVB	#G20,5.(R0)	:MAINTENANCE MODE
024332	112760	000020	000005		CLR	6.(R0)	:NO SEQUENCE #
024340	005060	000006			MOV	#RSMSIZ,R1	:SIZE OF PACKET
024344	012701	000012			TST	(R1)+	:PLUS FLAG+COUNT INTO R1
024350	005721				MOV	#RSSNSZ,JNDCNT(R5)	:SET UP SIZE TO SEND
024352	012765	000016	000070				
024360	004737	013654			CALL	CHKSUM	:FORM CHECKSUM R1=COUNT
024364	010110				MOV	R1,(R0)	:INSERT IN PACKET
024366	012701	001000			MOV	#512.,R1	:SET EXPECTATIONS:
024372	012703	000034					:CALC # OF DATA PACKETS TO EXPECT
024376	000503				MOV	#XSFLG,R3	:OFFSET OF FLAG
024400	005002				ADD	R5,R3	:ABS. ADDR. OF XSFLG
024402	005202			69\$:	CLR	R2	:PRESET
024404	012723	000001			INC	R2	:# PACKETS EXPECTED
024410	012723	000204			MOV	#RSDATA,(R3)+	:LOAD XSFLG
024414	162701	000200			MOV	#132.,(R3)+	:AND EXPECT COUNT
024420	101401				SUB	#128.,R1	:NEG RESULT LAST TIME
024422	000767				BLOS	71\$	:LAST TIME!
024424	005202			71\$:	BR	69\$	:MORE TO DO
024426	010265	000032			INC	R2	:ADD ONE FOR END PACK
024432	012723	000002			MOV	R2,XSPKMM(R5)	:SAVE # PACKETS TO EXPECT
024436	012713	000016			MOV	#RSEND,(R3)+	:EXPECT AN END
					MOV	#RSNDSZ,(R3)	:THIS BIG-14. BYTES
024442	004737	006560			CALL	RSVP	:SEND
							:AND RETURN TO SCHEDULER
3741	024452	005365	000066		DEC	TMP(R5)	:DO ALL RECORDS THIS TRACK?
3742	024456	001404			BEQ	2\$	:YES-GET OTHER TRACK
3743	024460	005265	000064		INC	REC(R5)	:NO-NEXT RECORD
3744	024464	000137	024030		JMP	1\$	:EXECUTE THE READ
3745	024470	005765	000062	2\$:	TST	TRK(R5)	:DONE 2 TRACKS?
3746	024474	001012			BNE	TST6EX	:YES-EXIT
3747	024476	005265	000062		INC	TRK(R5)	:NO-SET FLAG FOR NEXT PASS
3748	024502	013765	003334	000064	MOV	SECREC,REC(R5)	:GET NEW STARTING BLOCK #

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

L10021: TRAP C\$ETST

```

3757 .SBTTL TEST 7 / WRITE-VERIFY SELECTED NUMBER OF BLOCKS
3758
3759 BGNTST
3760 TSTID #TST7
3761 EXIT TST
3762
3763
3764 024576 005065 000064 000066 TST7: CLR REC(R5) ;START AT REC 0
3765 024602 013765 003310 000066 MOV TAPLEN,TMP(R5) ;GET THE # OF BLOCKS PER TRACK
3766 024610 005065 000062 000072 1$: CLR TRK(R5) ;TRK(R5)=1ST OR 2ND PASS
3767 024614 016565 000064 000072 1$: MOV REC(R5),PATTEN(R5) ;USE RECORD NO. FOR DATA
3768 024622 005737 002220 TST DRVCHK ;ADD DR #?
3769 024626 001403 BEQ 10$ ;NO
3770 024630 066565 000060 000072 10$: ADD DR(R5),PATTEN(R5) ;ADD DRIVE ID
3771 024636 012700 027722 72$: TUWRIT PATTEN(R5),REC(R5),#512.,DR(R5),#1
3772 024642 112710 000002 MOV #TRBUF,R0 ;MAKE COMMAND PACKET:
3773 024646 112760 000012 000001 MOV #RSCMD,R0 ;COMMAND FLAG
3774 024654 112760 000003 000002 MOV #RSMISZ,1(R0) ;THIS SIZE
3775 024662 112760 000001 000003 MOV #RSSW,2(R0) ;INSERT OP CODE-WRITE
3776 024670 116560 000060 000004 MOV #1,3.(R0) ;VERIFY (1 OR 0)
3777 024676 112760 000020 000005 MOV DR(R5),4.(R0) ;DRIVE #
3778 024704 005060 000006 000005 CLR #020,5.(R0) ;MAINTENANCE MODE SWITCH
3779 024710 012760 001000 000010 MOV 6.(R0) ;NO SEQUENCE #
3780 024716 016560 000064 000012 MOV #512.,8.(R0) ;TOTAL COUNT TO WRITE
3781 024724 012701 000012 MOV REC(R5),10.(R0) ;AT RECORD N
3782 024730 005721 TST #RSMISZ,R1 ;THE PACKET SIZE PLUS+2
3783 024732 012765 000016 000070 MOV (R1)+ ;((FLAG AND COUNT) INTO R1
3784 024740 004737 013654 CALL #RSSNSZ,SNDCNT(R5) ;LOAD THE SIZE TO SEND
3785 024744 010110 MOV CHKSUM ;R0 --> R1=COUNT
3786 024746 012765 000020 000034 MOV R1,(R0) ;PUT CHKSUM IN PACKET
3787 024754 012765 000001 000036 MOV #RSCONT,XSFLG(R5) ;SET UP EXPECTATIONS:
3788 024762 012765 000001 000032 MOV #1,XSCNT(R5) ;THE FLAG
3789 024770 012702 001000 MOV #1,XSPKNM(R5) ;THE COUNT
3790 024774 004737 006560 MOV #512.,R2 ;THE # PACKETS EXPECTED
3791 025000 032715 000010 CALL RSVP ;GET # OF DATA BYTES
3792 025004 001314 BIT #BIT3,R5 ;SEND (AND RETURN TO SCHEDULER)
3793 025006 042715 010000 BNE 72$ ;FLAG BYTE ERROR?
3794 025012 012700 027722 BIC #BIT12,R5 ;YES
3795 025016 020227 000200 MOV #TRBUF,R0 ;FLAG FOR LAST PACKET
3796 025022 101004 CMP R2,#128. ;POINT TO TOP OF BUFFER AGAIN
3797 025024 010201 BHI 65$ ;START DATA PACKET(S)
3798 025026 052715 010000 MOV R2,R1 ;#512. > 128.!
3799 025032 000402 BIS #BIT12,R5 ;#512.<128.
3800 66$ BR 66$ ;SO LAST PACKET NOW
3801 ;USE REMAINING COUNT

```

025034	012701	000200	
025040	110160	000001	
025044	010103		
025046	112710	000001	
025052	005720		
025054	116520	000072	
025060	005303		
025062	101374		
025064	012700	027722	
025070	116001	000001	
025074	042701	177400	
025100	010165	000070	
025104	062765	000004	000070
025112	062701	000002	
025116	004737	013654	
025122	110120		
025124	000301		
025126	110120		
025130	032715	010000	
025134	001412		
025136	012765	000002	000034
025144	012765	000016	000036
025152	012765	000001	000032
025160	000411		
025162	012765	000020	000034
025170	012765	000001	000036
025176	012765	000001	000032
025204	004737	006560	
025210	032715	000010	
025214	001210		
025216	032715	002000	
025222	001004		
025224	162702	000200	
025230	101270		
025232	000502		
025234			
025234	012700	027722	
025240	112710	000002	
025244	112760	000012	000001
025252	112760	000002	000002
025260	016560	000064	000012
025266	116560	000060	000004
025274	105060	000003	
025300	105715		
025302	100002		
025304	105260	000003	
025310	012760	001000	000010
025316	112760	000020	000005
025324	005060	000006	
025330	012701	000012	
025334	005721		
025336	012765	000016	000070
025344	004737	013654	

65\$:	MOV	#128.,R1	:USE 128. BYTES
66\$:	MOVB	R1,1(R0)	:COPY COUNT TO BUFFER
	MOV	R1,R3	:R3=COUNTER TO LOAD BUFFER
	MOVB	#RSDATA,@R0	:FLAG FIRST
	TST	(R0)+	:SKIP COUNT
67\$:	MOVB	PATTEN(R5),(R0)+	:INSERT DATA
	DEC	R3	:MORE?
	BHI	67\$	:YES
	MOV	#TRBUF,R0	:-->TOP AGAIN
	MOVB	1(R0),R1	:GET COUNT
	BIC	#177400,R1	:ZERO SIGN EXTEND
	MOV	R1,SND CNT(R5)	:HOW MANY TO SEND PLUS
	ADD	#4,SND CNT(R5)	:FLAG,COUNT,CHKSUM
	ADD	#2,R1	:COMPENSATE FOR FLAG + COUNT
	CALL	CHKSUM	:FOR CHECKSUM CALC.
	MOVB	R1,(R0)+	:CHKSUM INTO PACKET
	SWAB	R1	:EVEN ON AN ODD
	MOVB	R1,(R0)+	:BYTE BOUNDARY
	BIT	#BIT12,@R5	:LAST DATA PACKET?
	BEQ	68\$	:NO
	MOV	#RSEND,XSFLG(R5)	:YES-EXPECT 'END'
	MOV	#RSNDSZ,XSCNT(R5)	:OF THIS SIZE
	MOV	#1,XSPKNT(R5)	:AND 1 PACKET
	BR	69\$	:SEND
68\$:	MOV	#RSCONT,XSFLG(R5)	: (NOT LAST), EXPECT 'CONTINUE'
	MOV	#1,XSCNT(R5)	:AND 1 BYTE
	MOV	#1,XSPKNT(R5)	:AND 1 PACKET
69\$:	CALL	RSVP	:SEND PACKET
			:AND RETURN TO SCHEDULER
	BIT	#BIT3,@R5	:FLAG BYTE RETRY?
	BNE	72\$	:YES
	BIT	#BIT10,@R5	:RETRY DATA ERROR?
	BNE	70\$	:YES
	SUB	#128.,R2	:NO, MORE DATA TO SEND?
	BHI	64\$	:YES
	BR	71\$	:NO
70\$:	TURTRY	REC(R5),#512.,DR(R5)	:RETRY HERE
76\$:	MOV	#TRBUF,R0	:FORM CMD PACK:
	MOVB	#RSCMD,@R0	:MESSAGE PACK TYPE
	MOVB	#RSMISZ,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	77\$	:NO
	INCB	3(R0)	:YES-CHANGE THRESHOLD
77\$:	MOV	#512.,8.(R0)	:# BYTES DESIRED
	MOVB	#020,5.(R0)	:MAINTENANCE MODE
	CLR	6.(R0)	:NO SEQUENCE #
	MOV	#RSMISZ,R1	:SIZE OF PACKET
	TST	(R1)+	:PLUS FLAG+COUNT INTO R1
	MOV	#RSSNSZ,SND CNT(R5)	:SET UP SIZE TO SEND
	CALL	CHKSUM	:FORM CHECKSUM R1=COUNT

025350	010110				MOV	R1,(R0)	;INSERT IN PACKET
025352	012701	001000			MOV	#512.,R1	;SET EXPECTATIONS:
025356	012703	000034			MOV	#XSFLG,R3	;CALC # OF DATA PACKETS TO EXPECT
025362	060503				ADD	R5,R3	;OFFSET OF FLAG
025364	005002				CLR	R2	;ABS. ADDR. OF XSFLG
025366	005202				INC	R2	;PRESET
025370	012723	000001		73\$:	MOV	#RSDATA,(R3)+	;# PACKETS EXPECTED
025374	012723	000204			MOV	#132.,(R3)+	;LOAD XSFLG
025400	162701	000200			SUB	#128.,R1	;AND EXPECT COUNT
025404	101401				BLOS	75\$	;NEG RESULT LAST TIME
025406	000767				BR	73\$	;LAST TIME!
025410	005202			75\$:	INC	R2	;MORE TO DO
025412	010265	000032			MOV	R2,XSPKMM(R5)	;ADD ONE FOR END PACK
025416	012723	000002			MOV	#RSEND,(R3)+	;SAVE # PACKETS TO EXPECT
025422	012713	000016			MOV	#RSDSZ,(R3)	;EXPECT AN END
							;THIS BIG-14. BYTES
025426	004737	006560			CALL	RSVP	;SEND
							;AND RETURN TO SCHEDULER
3772	025442	005365	000066		DEC	TMP(R5)	;DO ALL RECORDS FOR THIS TRACK?
3773	025446	001404			BEQ	2\$	;YES-GET OTHER TRACK
3774	025450	005265	000064		INC	REC(R5)	;NO-NEXT RECORD
3775	025454	000137	024614		JMP	1\$	;EXECUTE THE WRITE
3776	025460	005765	000062	2\$:	TST	TRK(R5)	;DONE 2 TRACKS?
3777	025464	001012			BNE	TST7EX	;YES-EXIT
3778	025466	005265	000062		INC	TRK(R5)	;NO-SET FLAG FOR NEXT PASS
3779	025472	013765	003334	001064	MOV	SECREC,REC(R5)	;GET NEW STARTING BLOCK #
3780	025500	013765	003310	000066	MOV	TAPLEN,TMP(R5)	;RESET # OF BLOCKS
3781	025506	000137	024614		JMP	1\$	;AND EXECUTE
3782	025512	005237	003322		TST7EX: INC	DONE	;DONE
3783	025516	000207			RETURN		;RETURN
3784							
3785	025520				ENDTST		
	025520						
	025520	104401					
						L10022:	TRAP CSETST

```

3788 .SBTTL TEST 8 / READ-REDUCED THRESHOLD SELECTED NUMBER OF BLOCKS
3789
3790 025522 BGNTST
025522 T8::
3791 025522 TSTID #TST8
025522 012737 025566 003326 MOV #TST8,TSTTOP ;SAVE ADDR OF TEST
025530 004737 006004 CALL SETUP ;INIT UNITS TSTPC
025534 004737 005632 CALL SETDR ;GET 1ST DRVS.
025540 004737 006052 CALL RUN ;DO TEST
025544 004737 005530 CALL SWAPDR ;GET NEXT DRVS.
025550 103004 BCC 64$ ;BR NO 2ND DRVS
025552 004737 006004 CALL SETUP ;REINIT UNITS TSTPC
025556 004737 006052 CALL RUN ;REPEAT TEST
025562 ;DONE
3792 025562 EXIT TST
025562 104432
025564 000520 TRAP C$EXIT
;WORD L10023-.

3793
3794
3795 025566 005065 000064 TST8: CLR REC(R5) ;START AT REC 0
3796 025572 013765 003310 000066 MOV TAPLEN,TMP(R5) ;GET THE # OF BLOCKS PER TRACK
3797 025600 005065 000062 CLR TRK(R5) ;TRK(R5)=1ST OR 2ND PASS
3798 025604 016565 000064 000072 1$: MOV REC(R5),PATTEN(R5) ;USE RECORD NO. FOR DATA
3799 025612 005737 002220 TST DRVCHK ;ADD DR #?
3800 025616 001403 BEQ 10$ ;NO
3801 025620 066565 000060 000072 ADD DR(R5),PATTEN(R5) ;ADD DRIVE ID
3802 025626 10$: TUREAD REC(R5),#512.,DR(R5),#1

025626 012700 027722 68$: MOV #TRBUF,R0 ;FORM CMND PACK:
025632 112710 000002 MOV #RSCMND,R0 ;MESSAGE PACK TYPE
025636 112760 000012 000001 MOV #RSMISZ,1(R0) ;THIS BIG
025644 112760 000002 000002 MOV #RSSRD,2(R0) ;OP CODE IS READ
025652 016560 000064 000012 MOV REC(R5),10.(R0) ;THIS RECORD
025660 116560 000060 000004 MOV DR(R5),4.(R0) ;THIS DRIVE
025666 112760 000001 000003 MOV #1,3.(R0) ;VERIFY
025674 012760 001000 000010 MOV #512.,8.(R0) ;TOTAL BYTES TO READ
025702 112760 000020 000005 MOV #020.,5.(R0) ;MAINTENANCE MODE
025710 005060 000006 CLR 6.(R0) ;NO SEQUENCE #
025714 012701 000012 MOV #RSMISZ,R1 ;GET SIZE OF PACKET
025720 005721 TST (R1)+ ;+2 FOR CHECKSUM
025722 012765 000016 000070 MOV #RSSNSZ,SND CNT(R5) ;SIZE TO SEND
025730 004737 013654 CALL CHKSUM ;FORM CHECKSUM R1=COUNT
025734 010110 MOV R1,(R0) ;INSERT CHECKSUM

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

```


025776	010265	000032			MOV	R2,XSPKNM(R5)	:SAVE # PACKETS TO EXPECT
026002	012723	000002			MOV	#RSEND,(R3)+	:EXPECT AN END ALSO...
026006	012713	000016			MOV	#RSNDSZ,(R3)	:THIS BIG-14. BYTES
026012	004737	006560			CALL	RSVP	:SEND
							:AND RETURN TO SCHEDULER
026016	032715	002010		67\$:	BIT	#BIT10!BIT3,@R5	:RETRY?
026022	001500				BEQ	65\$	:NO.
026024					TURTRY	REC(R5),#512.,DR(R5)	:YES
026024	012700	027722		72\$:	MOV	#TRBUF,R0	:FORM CMND PACK:
026030	112710	000002			MOVB	#RSCMD,@R0	:MESSAGE PACK TYPE
026034	112760	000012	000001		MOV	#RSMISZ,1,(R0)	:THIS BIG
026042	112760	000002	000002		MOVB	#RSSRD,2,(R0)	:OP CODE-READ
026050	016560	000064	000012		MOV	REC(R5),10,(R0)	:THIS RECORD
026056	116560	000060	000004		MOVB	DR(R5),4,(R0)	:THIS DRIVE
026064	105060	000003			CLRB	3(R0)	:PRESET NORM THRESHOLD
026070	105715				TSTB	@R5	:REDUCED?
026072	100002				BPL	73\$	:NO
026074	105260	000003			INCB	3(R0)	:YES-CHANGE THRESHOLD
026100	012760	001000	000010	73\$:	MOV	#512.,8,(R0)	:# BYTES DESIRED
026106	112760	000020	000005		MOVB	#020,5,(R0)	:MAINTENANCE MODE
026114	005060	000006			CLR	6,(R0)	:NO SEQUENCE #
026120	012701	000012			MOV	#RSMISZ,R1	:SIZE OF PACKET
026124	005721				TST	(R1)+	:PLUS FLAG+COUNT INTO R1
026126	012765	000016	000070		MOV	#RSSNSZ,SND CNT(R5)	:SET UP SIZE TO SEND
026134	004737	013654			CALL	CHKSUM	:FORM CHECKSUM R1=COUNT
026140	010110				MOV	R1,(R0)	:INSERT IN PACKET
026142	012701	001000			MOV	#512.,R1	:SET EXPECTATIONS:
							:CALC # OF DATA PACKETS TO EXPECT
026146	012703	000034			MOV	#XSFLG,R3	:OFFSET OF FLAG
026152	060503				ADD	R5,R3	:ABS. ADDR. OF XSFLG
026154	005002				CLR	R2	:PRESET
026156	005202			69\$:	INC	R2	:# PACKETS EXPECTED
026160	012723	000001			MOV	#RSDATA,(R3)+	:LOAD XSFLG
026164	012723	000204			MOV	#132.,(R3)+	:AND EXPECT COUNT
026170	162701	000200			SUB	#128.,R1	:NEG RESULT LAST TIME
026177	101401				BLOS	71\$	:LAST TIME!
026177	000767				BR	69\$	:MORE TO DO
026200	005202			71\$:	INC	R2	:ADD ONE FOR END PACK
026212	010265	000032			MOV	R2,XSPKNM(R5)	:SAVE # PACKETS TO EXPECT
026212	012723	000002			MOV	#RSEND,(R3)+	:EXPECT AN END
026212	012713	000016			MOV	#RSNDSZ,(R3)	:THIS BIG-14. BYTES
026216	004737	006560			CALL	RSVP	:SEND
							:AND RETURN TO SCHEDULER
3803	026226	005365	000066		DEC	TMP(R5)	:DO ALL RECORDS THIS TRACK?
3804	026232	001404			BEQ	2\$	:YES-GET OTHER TRACK
3805	026234	005265	000064		INC	REC(R5)	:NO-NEXT RECORD
3806	026240	000137	025604		JMP	1\$	:EXECUTE THE READ
3807	026244	005765	000062	2\$:	TST	TRK(R5)	:DONE 2 TRACKS?
3808	026250	001012			BNE	TST8EX	:YES-EXIT
3809	026252	005265	000062		INC	TRK(R5)	:NO-SET FLAG FOR NEXT PASS
3810	026256	013765	000064		MOV	SECREC,REC(R5)	:GET NEW STARTING BLOCK #

MISCELLANEOUS SECTIONS MACRO M1113 25-FEB-82 07:06 PAGE 115-2
TEST 8 / READ-REDUCED THRESHOLD SELECTED NUMBER OF BLOCKS

SEQ 0116

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

L10023: TRAP C\$ETST


```

3819 .SBTTL TEST 9 / TESTS MODIFIED RADIAL SERIAL PROTOCOL
3820
3821 026306 BGNTST
3822 026306 T9::
3823 026306 012737 026330 003326 MOV #TST9,TSTTOP ;SAVE ADDR OF TEST
3824 026314 004737 006004 CALL SETUP ;INIT UNITS TSTPC
3825 026320 004737 006052 CALL RUN ;DO TEST
3826 ;DONE
3827
3828 026324 EXIT TST
3829 026324 104432 TRAP
3830 026326 000662 .WORD C$EXIT
3831 L10024-.
3830 026330 012737 000001 003342 TST9: MOV #1,TEST9 ;INDICATES 1ST PART OF TST 8
3831 026336 012700 027722 64$: MOV #TRBUF,R0 ;FORM COMMAND PACKET
3832 026342 112710 000002 MOV #RSCMND,R0 ;COMMAND FLAG
3833 026346 112760 000012 000001 MOV #RSMSIZ,1(R0) ;SIZE OF MESSAGE
3834 026354 112760 000012 000002 MOV #RSSGET,2(R0) ;GET CHARACTERISTICS
3835 026362 105060 000003 CLRB 3(R0) ;NO MODIFIER.
3836 026366 005060 000004 CLR 4(R0) ;NO DRIVE OR SWITCHES
3837 026372 005060 000006 CLR 6(R0) ;NO SEQUENCE NUMBER
3838 026376 005060 000010 CLR 8(R0) ;NO BYTES
3839 026402 005060 000012 CLR 10(R0) ;NO RECORD #
3840 026406 012701 000012 MOV #RSMSIZ,R1 ;GET SIZE
3841 026412 005721 TST (R1)+ ;+2 FOR CHECKSUM
3842 026414 012765 000016 000070 MOV #RSSNSZ,SND CNT(R5) ;SIZE TO SEND
3843 026422 004737 013654 CALL CHKSUM ;FORM CHECKSUM
3844 026426 010110 MOV R1,(R0) ;INSERT INTO PACKET
3845 026430 012765 000001 000034 MOV #RSDATA,XSFLG(R5) ;EXPECT DATA PACKET
3846 026436 012765 000034 000036 MOV #RSGCDP,XSCNT(R5) ;THIS BIG
3847 026444 012765 000001 000032 MOV #1,XSPKNT(R5) ;AND 1 PACKET
3848 ;SEND
3849 026452 004737 006560 CALL RSVP ;RETURN TO SCHEDULER
3850
3851 026456 004737 014010 CALL DOBRK ;CLR POTENTIAL INTERFACE ERROR
3852
3853 026462 032715 000010 BIT #BIT3,R5 ;RETRY?(BAD FLAG)
3854 026466 001323 BNE 64$ ;YES
3855
3856 026470 012737 000002 003342 MOV #2,TEST9 ;INDICATE 2ND PART OF TST 8
3857
3858 026476 012700 027722 65$: MOV #TRBUF,R0 ;-->(POINT TO) XMIT BUFFER
3859 026502 112710 000002 MOV #RSCMND,R0 ;FORM COMMAND MESSAGE PACK
3860 026506 112760 000012 000001 MOV #RSMSIZ,1(R0) ;THIS BIG
3861 026514 112760 000001 000002 MOV #RSSNIT,2(R0) ;OP CODE IS INITIALIZE
3862 026522 013760 000064 000012 MOV REC,10.(R0) ;TO THIS RECORD
3863 026530 105060 000003 CLRB 3.(R0) ;NO MODIFIER
3864 026534 105060 000004 CLRB 4.(R0) ;NO DRIVE
3865 026540 112760 000010 000005 MOV #BIT03,5.(R0) ;SET MRSP SWITCH
3866 026546 005060 000006 CLR 6.(R0) ;NO SEQUENCE #
3867 026552 005060 000010 CLR 8.(R0) ;NO BYTE COUNT
3868 026556 012701 000012 MOV #RSMSIZ,R1 ;GET COUNT
3869 026562 005721 TST (R1)+ ;PLUS FLAG + BCNT
3870 ;FOR CHECKSUM CALC
3871 026564 004737 013654 CALL CHKSUM ;R0-->TOP R1=# OF BYTES
3872 026570 010110 MOV R1,(R0) ;INSERT INTO PACKET

```

```

3873                                     :SET UP EXPECTATIONS:
3874 026572 012765 000016 000070      MOV  #RSSNSZ,SND CNT(R5)  :HOW MANY TO SEND
3875 026600 112765 000002 000034      MOV  #RSCMND,XSFLG(R5)  :EXPECT END PACK
3876 026606 012765 000016 000036      MOV  #RSNDSZ,XSCNT(R5)  :COUNT WITH THIS
3877 026614 012765 000001 000032      MOV  #1.,XSPKNM(R5)  :EXPECT ONLY 1 PACKET
3878
3879 026622 004737 006560      CALL  RSVP          :SEND
3880                                     :AND RETURN TO SCHEDULER
3881
3882 026626 032715 000010      BIT   #BIT3,AR5      :RETRY (FLAG BYTE ERROR)?
3883 026632 001321                BNE   66$          :YES
3884
3885 026634 012700 027722      66$:  MOV  #TRBUF,R0      :-->(POINT TO) XMIT BUFFER
3886 026640 112710 000002      MOV  #RSCMND,AR0      :FORM COMMAND MESSAGE PACK
3887 026644 112760 000012 000001      MOV  #RSMSIZ,1(R0)  :THIS BIG
3888 026652 112760 000000 000002      MOV  #RSSNOP,2(R0)  :OP CODE IS NO-OPERATION
3889 026660 013760 000064 000012      MOV  REC,10.(R0)  :TO THIS RECORD
3890 026666 105060 000003      CLRB  3.(R0)          :NO MODIFIER
3891 026672 105060 000004      CLRB  4.(R0)          :NO DRIVE
3892 026676 112760 000010 000005      MOV  #BIT03,5.(R0)  :SET MRSP SWITCH
3893 026704 005060 000006      CLR   6.(R0)          :NO SEQUENCE #
3894 026710 005060 000010      CLR   8.(R0)          :NO BYTE COUNT
3895 026714 012701 000012      MOV  #RSMSIZ,R1      :GET COUNT
3896 026720 005721                TST   (R1)+        :PLUS FLAG + BCNT
3897                                     :FOR CHECKSUM CALC
3898 026722 004737 013654      CALL  CHKSUM          :RO-->TOP  R1=# OF BYTES
3899 026726 010110      MOV  R1,(R0)              :INSERT INTO PACKET
3900                                     :SET UP EXPECTATIONS:
3901 026730 012765 000016 000070      MOV  #RSSNSZ,SND CNT(R5)  :HOW MANY TO SEND
3902 026736 112765 000002 000034      MOV  #RSCMND,XSFLG(R5)  :EXPECT END PACK
3903 026744 012765 000016 000036      MOV  #RSNDSZ,XSCNT(R5)  :COUNT WITH THIS
3904 026752 012765 000001 000032      MOV  #1.,XSPKNM(R5)  :EXPECT ONLY 1 PACKET
3905
3906 026760 004737 006560      CALL  RSVP          :SEND
3907                                     :AND RETURN TO SCHEDULER
3908
3909 026764 032715 000010      BIT   #BIT3,AR5      :RETRY (FLAG BYTE ERROR)?
3910 026770 001321                BNE   66$          :YES
3911
3912 026772 005237 003322      INC    DONE
3913 026776 005037 003342      CLR    TEST9
3914
3915 027002 005737 002224      TST   PPSOT9          :PROTOCOL SUMMARY @ END OF PASS
3916 027006 00147                BEQ    ENDT9          :NO
3917 027010 005037 027366      CLR    UNITNO        :SET UNIT # TO ZERO
3918 027014                PRINTF  #MSAGE1          :PRINT HEADER
3919 027014 012746 027212      MOV  #MSAGE1,-(SP)
3920 027020 012746 000001      MOV  #1,-(SP)
3921 027024 010600                MOV  SP,R0
3922 027026 104417                TRAP  C$PNTF
3923 027030 062706 000004      ADD   #4,SP
3924 027034 012737 003350 003312      MOV  #BLKTBL,DEV PTR  :SET ALL UNITS
3925 027042 017705 154244      MOV  @DEV PTR,R5      :GET POINTER
3926 027046 005765 000000      TST   STATUS(R5)      :IS UNIT ABORTED
3927 027052 100431                BMI   3$            :YES
3928 027054 005765 000210      TST   MRSP(R5)        :IS UNIT MODIFIED
3929 027060 001413                BEQ    2$            :NO

```

```

3925 027062          PRINTF #MSAGE2,UNITNO ;MESSAGE FOR MODIFIED UNIT
      027062 013746 027366
      027066 012746 027253
      027072 012746 000002
      027076 010600
      027100 104417
      027102 062706 000006
3926 027106 000425
3927 027110          2$: BR 4$ ;SEE IF LAST UNIT
      027110 013746 027366          PRINTF #MSAGE3,UNITNO ;MESSAGE FOR NON-MODIFIED UNIT
      027114 012746 027307
      027120 012746 000002
      027124 010600
      027126 104417
      027130 062706 000006
3928 027134 000412          3$: BR 4$ ;SEE IF LAST UNIT
3929 027136          PRINTF #MSAGE4,UNITNO ;MESSAGE FOR ABORTED UNIT
      027136 013746 027366
      027142 012746 027336
      027146 012746 000002
      027152 010600
      027154 104417
      027156 062706 000006
3930 027162 023727 003312 003366 4$: CMP DEVPTR,#LSTDEV ;IS THIS THE LAST DEVICE
3931 027170 103006          BHIS ENDT9 ;YES
3932 027172 062737 000002 003312 ADD #2,DEVPTR ;GET NEXT UNIT
3933 027200 005237 027366 INC UNITNO ;INC UNIT #
3934 027204 000716 BR 1$
3935
3936 027206 000207          ENDT9: RETURN
3937
3938 027210          ENDTST
      027210
      027210 104401
3939
3940 027212 045 116 045 MSAGE1: .ASCIZ /%N%S8%UNIT NO%S9%S6%APROTOCOL%N/
3941 027253 045 116 045 MSAGE2: .ASCIZ !%N%S9%S2%01%S9%S9%ARSP/MRSP!
3942 027307 045 116 045 MSAGE3: .ASCIZ /%N%S9%S2%01%S9%S9%ARSP/
3943 027336 045 116 045 MSAGE4: .ASCIZ /%N%S9%S2%01%S9%S9%A---/
3944 .EVEN
3945 027366 000000 UNITNO: .WORD

```

L10024: TRAP CSETST

MISCELLANEOUS SECTIONS MACRO M1113 25-FEB-82 07:06 PAGE 119
PATCH AREA

SEQ 0120

3948
3949
3950
3951

000144

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

3954
3955
3956
3957
3958 027700 030760
3959 027702 032016
3960 027704 033054
3961 027706 034112
3962 027710 035150
3963 027712 036206
3964 027714 037244
3965 027716 040302
3966
3967
3968
3969
3970
3971 027720 023
3972 027721 023
3973
3974 027722
3975
3976
3977
3978 030760
3979 032016
3980 033054
3981 034112
3982 035150
3983 036206
3984 037244
3985 040302
3986
3987
3988
3989 041340

.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

PARAMETER CODING
I/O BUFFER AREAS:

MACRO M1113 25-FEB-82 07:06 PAGE 123

SEQ 0122

4013
4024
4025
4053
4054 041340
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065 041340
041340 000021
041342
4066
4067
4068 041342
041342 000031
041344 041404
041346 160000
041350 177777
4069 041352
041352 001031
041354 041415
041356 000000
041360 000776
4070 041362
041362 003130
041364 041432
041366 000001
4071 041370
041370 002130
041372 041450
041374 000001
4072 041376
041376 002130
041400 041465
041402 000002
4073
4079
4080 041404
041404
4081
4082 041404 124 125 065
4083 041415 126 105 103
4084 041432 120 104 124
4085 041450 124 105 123
4086 041465 124 105 123
4087
4088
4089

.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

.WORD L10025-L\$HARD/2
L\$HARD::

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

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

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

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

GPRML MSG1C,6,1,YES

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

GPRML MSG2,4,1,YES

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

GPRML MSG3,4,2,YES

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

ENDHRD

.EVEN
L10025:

MSG1: .ASCIZ /TU58 CSR/
MSG1A: .ASCIZ /VECTOR ADDR./
MSG1C: .ASCIZ /PDT INTERFACE/
MSG2: .ASCIZ /TEST DRIVE 0/
MSG3: .ASCIZ /TEST DRIVE 1/
.EVEN

4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108

.SBTTL SOFTWARE PARAMETER CODING SECTION

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

041502
041502 00003
041504
4109 041504
041504 000052
041506 041566
041510 001777
041512 000010
041514 001000
4110 041516
041516 004130
041520 041633
041522 000001
4111 041524
041524 001130
041526 041675
041530 000001
4112 041532
041532 003130
041534 041727
041536 000001
4113 041540
041540 002130
041542 041754
041544 000001
4114 041546
041546 005052
041550 042002
041552 000377
041554 000001
041556 000376
4115 041560
041560 006130
041562 042043
041564 000001
4122 041566

BGNSFT

GPRMD MSG4,0,D,1777,8.,512.,YES

GPRML MSG4B,10,1,YES

GPRML MSG5,2,1,YES

GPRML MSG6,6,1,YES

GPRML MSG7,4,1,YES

GPRMD MSG8,10.,D,377,1,254.,YES

GPRML MSG9,12.,1,YES

SFTOUT: ENDSFT

.WORD L10026-L\$SOFT/2
L\$SOFT::

.WORD T\$CODE
.WORD MSG4
.WORD 1777
.WORD T\$LOLIM
.WORD T\$HILIM

.WORD T\$CODE
.WORD MSG4B
.WORD 1

.WORD T\$CODE
.WORD MSG5
.WORD 1

.WORD T\$CODE
.WORD MSG6
.WORD 1

.WORD T\$CODE
.WORD MSG7
.WORD 1

.WORD T\$CODE
.WORD MSG8
.WORD 377
.WORD T\$LOLIM
.WORD T\$HILIM

.WORD T\$CODE
.WORD MSG9
.WORD 1

.EVEN

L10026:

041566
4123 041566 116 125 115 MSG4: .ASCIIZ 'NUMBER OF BLOCKS:TEST 5-8 (8 TO 512)'
4124 041633 101 104 104 MSG4B: .ASCIIZ /ADD DR # TO DATA PATTERN:TEST 5-8/
4125 041675 123 124 101 MSG5: .ASCIIZ /STATISTICS PRINTED AT EOP/
4126 041727 103 117 115 MSG6: .ASCIIZ /COMPARE DATA ON READ/
4127 041754 120 122 111 MSG7: .ASCIIZ /PRINT PACKET ON ERROR/
4128 042002 043 040 105 MSG8: .ASCIIZ /# ERRORS = DVC FATAL IF 'EVL'SET/
4129 042043 120 122 111 MSG9: .ASCIIZ /PRINT UNIT PROTOCOL SUMMARY (TEST 9)/
4130 .EVEN

LINE	ADDRESS	DATA	OPERATION	REPEAT	REMARKS
4133	000016		.REPT	14.	;LASTAD CORRECTION
4134			.WORD		
4135			.ENDR		
4142	042144		LASTAD		
	042144	042164			
	042146	000006			
	042150		L\$LAST::		
4143	042150		ENDMOD		
4144					
4145	042150		BGNSETUP	1	
4146	042150		BGNPTAB		
	042150	000000			
	042152	000004			
	042154				
4147	042154	176500			
4148	042156	000300			
4149	042160	000003			
4150	042162	000000			
4151	042164		ENDPTAB		
	042164				
4152	042164		ENDSETUP		
4153		000001	.END		

.EVEN
.WORD TSFREE
.WORD TSSIZE

.WORD 0
.WORD L10031-./2-1

L10027:

L10031:

PARAMETER CODING
SYMBOL TABLE

MACRO M1113 25-FEB-82 07:06 PAGE 127-1

SEQ 0125

ABNDX = 000004 G	CHKANS 010576 G	C\$MSG = 000023	ESSLF = 177777 G	G\$OF SI = 000376
ABO 013146	CHKEND 011302 G	C\$OPEN= 000034	ESTRY = 000001 G	G\$PRMA= 000001
ABOMSG 017254	CHKERR 011406	C\$PNTB= 000014	ESWLOC= 177765 G	G\$PRMD= 000002
ABONM 006532	CHKPKS 010706 G	C\$PNTF= 000017	ESWR = 177757	G\$PRML= 000000
ADR = 000020 G	CHKPTR 010704	C\$PNTS= 000016	EVL = 000004 G	G\$RADA= 000140
ALLGON 003340 G	CHKREE 011460	C\$PNTX= 000015	EVLTHR 002222	G\$RADB= 000000
ASSEMB= 000010	CHKRET 012024	C\$QIO = 000377	EXOFF 007427	G\$RADD= 000040
BDATA = 000134 G	CHKSUC 012320 G	C\$RDBU= 000007	EXON 007426	G\$RADL= 000120
BDBYTS 014726	CHKSUM 013654 G	C\$REFG= 000047	ESEND = 002100	G\$RADO= 000020
BDCHK = 000022 G	CHK8 010616	C\$RESE= 000033	ESLOAD= 000035	G\$XFER= 000004
BDCOM = 000014 G	CKCKSM 013750 G	C\$REVI= 000003	FLGLOC 016752 G	G\$YES = 000010
BIT0 = 000001 G	CLRALL 005710 G	C\$RFLA= 000021	FM 015730	HARDR = 000136 G
BIT00 = 000001 G	CLRBUF 005750 G	C\$RPT = 000025	FMO 015712	HARDW = 000140 G
BIT01 = 000002 G	CLRPTR 006002	C\$SEFG= 000046	FTLNM 003320	HELP = 000000
BIT02 = 000004 G	CMDSNT= 000100 G	C\$SPRI= 000041	FSAU = 000015	HOE = 100000 G
BIT03 = 000010 G	CMNDER= 000040 G	C\$SVEC= 000037	FSAUTO= 000020	HRD 013126
BIT04 = 000020 G	CMPCAT 002216	C\$TPRI= 000013	F\$BGN = 000040	HRDRD = 000016 G
BIT05 = 000040 G	CNINIT= 000032 G	DESC 014730	F\$CLEA= 000007	HRDWR = 000020 G
BIT06 = 000100 G	COMPAR 014550 G	DEVPTTR 003312 G	F\$DU = 000016	HRD1 011746
BIT07 = 000200 G	CSNRDY 003344 G	DEV0 003370	F\$END = 000041	IBE = 010000 G
BIT08 = 000400 G	CSRCVB 003346 G	DEV1 003602	F\$HARD= 000004	IDPTR 003324 G
BIT09 = 001000 G	C\$AU = 000052	DEV2 004014	F\$HW = 000013	IDU = 000040 G
BIT1 = 000002 G	C\$AUTO= 000061	DEV3 004226	F\$INIT= 000006	IER = 020000 G
BIT10 = 002000 G	C\$BRK = 000022	DEV4 004440	F\$JMP = 000050	INIT 016164
BIT11 = 004000 G	C\$BSEG= 000004	DEV5 004652	F\$MOD = 000000	INITWD 014542
BIT12 = 010000 G	C\$BSUB= 000002	DEV6 005064	F\$MSG = 000011	INIT2 016212
BIT13 = 020000 G	C\$CEFG= 000045	DEV7 005276	F\$PROT= 000021	ISR = 000100 G
BIT14 = 040000 G	C\$CLCK= 000062	DFPTBL 002176 G	F\$PWR = 000017	IXE = 004000 G
BIT15 = 100000 G	C\$CLEA= 000012	DFTL1 012774	F\$RPT = 000012	ISAU = 000041
BIT2 = 000004 G	C\$CLOS= 000035	DIAGMC= 000000	F\$SEG = 000003	ISAUTO= 000041
BIT3 = 000010 G	C\$CLP1= 000006	DLV = 000074 G	F\$SOFT= 000005	ISCLN = 000041
BIT4 = 000020 G	C\$CVEC= 000036	DOBRK 014010 G	F\$SRV = 000010	ISDU = 000041
BIT5 = 000040 G	C\$DCLN= 000044	DONE 003322 G	F\$SUB = 000002	ISHRD = 000041
BIT6 = 000100 G	C\$DODU= 000051	DR = 000050 G	F\$SW = 000014	ISINIT= 000041
BIT7 = 000200 G	C\$DRPT= 000024	DRVCHK 002220	F\$TEST= 000001	ISMOD = 000041
BIT8 = 000400 G	C\$DU = 000053	EF.CON= 000036 G	GBTMP 010572	ISMSG = 000041
BIT9 = 001000 G	C\$EDIT= 000003	EF.NEW= 000035 G	GBTMP2 010574	ISPROT= 000040
BLKEND= 000202 G	C\$ERDF= 000055	EF.PWR= 000034 G	GETANS 007152 G	ISPTAB= 000041
BLKER 003332 G	C\$ERHR= 000056	EF.RES= 000037 G	GETHRD 016316	ISPR = 000041
BLKSIZ= 000212 G	C\$ERRO= 000060	EF.STA= 000040 G	GETPTR 007216	ISRPT = 000041
BLKTBL 003350 G	C\$ERSF= 000054	ENDGPB 010146	GETRS 017222	ISSEG = 000041
BOE = 000400 G	C\$ERSO= 000057	ENDRSP 007100	GTAGIN 007464	ISSETU= 000041
BRKPTR 014546	C\$ESCA= 000010	ENDT9 027206	GTBYTE 010346 G	ISSFT = 000041
BRKTO 014544	C\$ESEG= 000005	ERRDES 013170 G	GTDOWN 010104	ISSRV = 000041
BRKWD 014540	C\$ESUB= 000003	ERRMOD 010150	GTOK 007730	ISSUB = 000041
BUFTBL 027700	C\$ETST= 000001	ESABO = 177720 G	GTPKS1 007220 G	ISTST = 000041
BUF0 030760	C\$EXIT= 000032	ESCKS = 177757 G	GTPKS8 007430 G	JSJMP = 000167
BUF1 032016	C\$GETE= 000026	ESCKSM= 177757	GTPTR 010344	LENGTH 002210
BUF2 033054	C\$GETW= 000027	ESCMD = 177720 G	GTUM 007672	LGOFST= 000120 G
BUF3 034112	C\$GMAN= 000043	ESNCRT= 177767 G	G\$CNT0= 000200	LKCNF 015130
BUF4 035150	C\$GPHR= 000042	ESNOMO= 177737 G	G\$DELM= 000372	LOE = 040000 G
BUF5 036206	C\$GPLO= 000030	ESNONX= 177770 G	G\$DISP= 000003	LOG 012634 G
BUF6 037244	C\$GPRI= 000040	ESOK = 000000 G	G\$EXCP= 000400	LOGO 013156
BUF7 040302	C\$INIT= 000011	ESPART= 177776 G	G\$HILI= 000002	LOGOK 012722
CARLF 015144	C\$INLP= 000020	ESRD = 177757	G\$LOLI= 000001	LOGOK2 013006
CHECK 016274	C\$MANI= 000050	ESREC = 177711 G	G\$NO = 000000	LOGO 012710
CHKANR 010702	C\$MEM = 000031	ESSK = 177740 G	G\$OFFS= 000400	LOG1 013024

PARAMETER CODING
SYMBOL TABLE

MACRO M1113 25-FEB-82 07:06 PAGE 127-2

SEQ 0126

LOG2	013054	LSUNIT	002012	G	MSNRSP	002764	G	PRI07 =	000340	G	SETUP	006004	G
LOG3	013114	L10001	002206		MSOVRN	003260	G	PRNPAK	014764	G	SFPTBL	002210	G
LOG3B	013136	L10002	002226		MSPART	002634	G	PRNSIZ	003336	G	SFT	013104	
LOT =	000010	L10003	013360		MSQRSP	003000	G	PTR	017252		SFTOUT	041566	
LSTDEV	003366	L10004	014446		MSREC	002720	G	RCBCNT	003316		SFTRD =	000002	G
LSACP	002110	L10005	014502		MSRNIT	002536	G	RCBFSZ=	001036	G	SFTWR =	000004	G
LSAPT	002036	L10006	015572		MSELFF	002364	G	RCDB =	000024	G	SKERR =	000024	G
LSAU	017302	L10007	016662		MSSFRD	003054	G	RCFLG	003314	G	SLFER =	000044	G
LSAUT	002070	L10010	017060		MSSFWR	003114	G	RCINIT=	000006	G	SND	006650	
LSAUTO	016754	L10011	017154		MSSKER	002306	G	RCSR =	000022	G	SNDBYT	007102	G
LSCCP	002106	L10012	017220		MSTOSN	003032	G	RCVBUF=	000102	G	SNDCNT=	000070	G
LSCLEA	017136	L10013	017302		MSUNIT	002656	G	RCVHND	014450		SNDHND	014434	
LSCO	002032	L10014	017504		MSWPRO	002514	G	RCVINT	014450	G	SNDINT	014434	G
LSDEPO	002011	L10015	017756		MSWRSP	002740	G	RDNO =	000114	G	SOFT R =	000122	G
LSDESC	002122	L10016	021350		MXRTRY	003330	G	RDNI =	000116	G	SOFT W =	000124	G
LSDESP	002076	L10017	022754		NCART =	000054	G	REC =	000064	G	SRVTBL	010322	
LSDEVP	002060	L10020	023744		NODRVS	016714		RECDAT	017742		STAEOP	002212	
LSDISP	002152	L10021	024530		NOMOR	006534		RECERR=	000042	G	STATHD	015600	
LSDLY	002116	L10022	025520		NOMOT =	000030	G	RECID	013442	G	STATUS=	000000	G
LSDTP	002040	L10023	026304		NOREE	011334		RECID2	013616		STHD2	016054	
LSDTYP	002034	L10024	027210		NOUNIT=	000036	G	RECOV	012044		STRT	016746	G
LSDU	017156	L10025	041404		NOXOFF	006632		RETErr	012246		SUCCS =	000076	G
LSDUT	002072	L10026	041566		NTSFT	013036		RETRY =	000002	G	SUCOTL=	000046	G
LSDVTY	005510	L10027	042154		NXTRET	006530		RLUN	015574		SVCGBL=	000000	
LSEF	002052	L10031	042164		NXTST	006102	G	RPTR	015576		SVCINS=	000001	
LSENVI	002044	MABEE	013066		NXTST2	006352		RSCMND=	000002	G	SVC SUB=	000001	
LSETP	002102	MESMRS	010200		ONEFIL=	000001		RSCONT=	000020	G	SVCTAG=	000001	
LSEXP1	002046	MODRSP	010252		OTL =	000010	G	RSDASZ=	000204	G	SVCTST=	000001	
LSEXP4	002064	MRSPLY	010254		OVRFLD	013534		RSDATA=	000001	G	SWAPDR	005530	G
LSEXP5	002066	MRSP =	000210	G	OVN =	000012	G	RSDNSZ=	000222	G	SWPTR	005630	
LSHARD	041342	MSAGE1	027212		OSAPTS=	000000		RSEND =	000002	G	SYSTAT	003306	G
LSHIME	002120	MSAGE2	027253		OSAU =	000001		RSGCDP=	000034	G	SSLSYM=	010000	
LSHPCP	002016	MSAGE3	027307		OSBGNR=	000001		RSINIT=	000004	G	TAPLEN	003310	G
LSHPTP	002022	MSAGE4	027336		OSBGNS=	000001		RSMSIZ=	000012	G	TEST9	003342	G
LSHW	002175	MSAUTO	017116		OSDU =	000001		RSNDZ=	000016	G	THPSHI	012152	
LSICP	002104	MSBDA	002340	G	OSERRT=	000000		RSNTAB	002226		THRSLO	012124	
LSINIT	016164	MSCMD	002704	G	OSGNSW=	000001		RSSEND=	000100	G	TMP =	000066	G
LSLADP	002026	MSCOM	002404	G	OSPOIN=	000001		RSSGET=	000012	G	TOMANY	016664	
LSLAST	042150	MSG1	041404		OSSETU=	000001		RSSNIT=	000001	G	TORCVB=	000050	G
LSLOAD	002100	MSG1B	041415		PARTL =	000034	G	RSSNOP=	000000	G	TOSNDB=	000056	G
LSLUN	002074	MSG1C	041432		PATTEN=	000072	G	RSSNSZ=	000016	G	TRBUF	027722	
LSMREV	002050	MSG2	041450		PDTFLG	016750	G	RSSRD =	000002	G	TRK =	000062	G
LSNAME	002000	MSG3	041465		PERDEV	006364		RSSSEK=	000005	G	TRPHND	017064	
LSPRIO	002042	MSG4	041566		PKPTR =	000104	G	RSSSLF=	000007	G	TRPPTR	017062	
LSPROT	002142	MSG4B	041633		PNT =	001000	G	RSSWR =	000003	G	TSTPC =	000020	G
LSPT	002112	MSG5	041675		PPSOT9	002224		RSVP	006560	G	TSTTOP	003326	
LSREPP	002062	MSG6	041727		PRBUF	002214		RSXOFF=	000023	G	TST1	017350	
LSREV	002010	MSG7	041754		PRDAT	015132		RSXON =	000020	G	TST2	017552	
LSRPT	015150	MSG8	042002		PRFORM	015134		RTRYN	012204		TST3	020024	
LSOFT	041504	MSG9	042043		PRI =	002000	G	RUN	006052	G	TST3PT	022742	
LSGPC	002056	MSHCHK	002556	G	PRI00 =	000000	G	SAVCNT=	000206	G	TST4	021416	
LSGPCP	002020	MSHDRD	003154	G	PRI01 =	000040	G	SECREC	003334	G	TST5	023022	
LSPTP	002024	MSHDWR	003216	G	PRI02 =	000100	G	SERVST	010342		TST5EX	023736	
LSSTA	002030	MSNIT	002620	G	PRI03 =	000140	G	SETDR	005632	G	TST6	024012	
LSW	002210	MSNLOG	002322	G	PRI04 =	000200	G	SETLEN	016624		TST6EX	024522	
LSTEST	002114	MSNOMO	002446	G	PRI05 =	000240	G	SETPTR	005706		TST7	024576	
LSTIML	002014	MSNOTP	002464	G	PRI06 =	000300	G	SETSRV	010256		TST7EX	025512	

PARAMETER CODING
SYMBOL TABLE

MACRO M1113 25-FEB-82 07:06 PAGE 127-3

K 10

SEQ 0127

TST8 025566
TST8EX 026276
TST9 026330
TUVECT= 000204 G
TSARGC= 000002
TSCODE= 006130
TSERRN= 000146
TSEXCP= 000000
TSFLAG= 000040
TSFREE= 042164
TSGMAN= 000000
TSHILI= 000376
TSLAST= 000001
TSLOLI= 000001
TSLSYM= 010000
TSLTNO= 000011
T\$NEST= 177777

T\$NSO = 000000
T\$NS1 = 000005
T\$PCNT= 000000
T\$PTAB= 010030
T\$PTHV= 000001
T\$PTNU= 000001
T\$SAVL= 177777
T\$SEGL= 177777
T\$SIZE = 000006
T\$SUBN= 000000
T\$TAGL= 177777
T\$TAGN= 010032
T\$TEMP= 000000
T\$TEST= 000011
T\$TSTM= 177777
T\$TSTS= 000001
T\$SAU = 010013

T\$\$AUT= 010010
T\$\$CLE= 010011
T\$\$DAT= 010031
T\$\$DU = 010012
T\$\$HAR= 010025
T\$\$HW = 010001
T\$\$INI= 010007
T\$\$MSG= 010003
T\$\$PC = 000001
T\$\$PRO= 010000
T\$\$PTA= 010030
T\$\$RPT= 010006
T\$\$SOF= 010026
T\$\$SRV= 010005
T\$\$SW = 010002
T\$\$TES= 010024
T1 017304 G

T1TRY = 000146 G
T2 017506 G
T3 017760 G
T4 021352 G
T4TRY = 000132 G
T5 022756 G
T6 023746 G
T7 024532 G
T8 025522 G
T9 026306 G
UAM = 000200 G
UNIT 013362 G
UNITNO 027366
UNREC 012224
UNSUC 011622
UNXPCT 007634

WAIT 014504
WHCHDR 013640 G
WRLOCK= 000026 G
WRTNO = 000110 G
WRTN1 = 000112 G
XFNSND 006640
XMDB = 000030 G
XMSR = 000026 G
XSCNT = 000036 G
XSFLG = 000034 G
XSPKMM= 000032 G
XSPTR = 000106 G
XSALWA= 000000
XSFALS= 000040
X\$OFFS= 000400
X\$TRUE= 000020

. ABS. 042164 000
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 33938 WORDS (133 PAGES)

DYNAMIC MEMORY: 21558 WORDS (82 PAGES)

ELAPSED TIME: 00:05:18

CZTUUDO.BIN/EN:AMA:ABS,CZTUUDO/CR/-SP=LB1:[1,1]SVC/MLB.SY:[203,377]CZTUUDO

CZTUJDO CREATED BY MACRO ON 25-FEB-82 AT 07:08 PAGE 1
 SYMBOL CROSS REFERENCE CREF
 SYMBOL VALUE REFERENCES

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

CZTUUD0 CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 2
CREF

SEQ 0129

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES	*82-2915	*82-2921	*82-2931	*82-2939	*84-3003	#84-3012
BRKTO		014544	*82-2906						
BRKWD		014540	82-2920	#84-3009					
BUFTBL		027700	91-3252	#121-3958					
BUFO		030760	121-3958	#121-3978					
BUF1		032016	121-3959	#121-3979					
BUF2		033054	121-3960	#121-3980					
BUF3		034112	121-3961	#121-3981					
BUF4		035150	121-3962	#121-3982					
BUF5		036206	121-3963	#121-3983					
BUF6		037244	121-3964	#121-3984					
BUF7		040302	121-3965	#121-3985					
CARLF		015144	88-3091	88-3093	#88-3102				
CHECK		016274	91-3255	#91-3260					
CHKANR		010702	64-2330	64-2342	#64-2346				
CHKANS	G	010576	47-1751	#64-2324					
CHKEND	G	011302	66-2402	66-2450	#68-2487				
CHKERR		011406	68-2500	#68-2507					
CHKPKS	G	010706	64-2328	64-2340	#66-2376				
CHKPTR		010704	*64-2332	64-2333	64-2341	*64-2343	#64-2348		
CHKREE		011460	68-2493	#68-2514					
CHKRET		012024	68-2498	68-2503	68-2506	68-2509	68-2513	68-2531	68-2539
			#68-2561						
CHKSUC	G	012320	68-2495	68-2514	#70-2596				
CHKSUM	G	013654	#78-2816	80-2859	102-3582	104-3600	106-3631	106-3631	106-3632
			107-3666	107-3666	107-3666	107-3667	107-3667	109-3709	109-3709
			111-3740	113-3771	113-3771	113-3771	115-3802	115-3802	117-3843
			64-2327	#64-2332					
CHK8		010616	66-2395	66-2433	66-2445	#80-2856			
CKCKSM	G	013750	#41-163	54-1988					
CLRALL	G	005710	41-1735	#43-1701	54-1991				
CLRBUF	G	005750	*41-1683	41-1684	41-1686	*41-1688	#43-1711		
CLRPTR		006002	#16-926	*50-1911	70-2603	70-2607	70-2620	74-2769	*82-2908
CMDSNT	=	000100	#11-688	70-2641					
CMNDER	=	000040	#9-579	86-3035					
CMPCAT		002216	#11-685	82-2961					
CNINIT	=	000032	66-2411	66-2435	#86-3030				
COMPAR	G	014550	#15-877	52-1956					
CSNRDY	G	003344	#15-878	62-2262					
CSRCVB	G	003346	#5-464	99-3513					
C\$AU	=	000052	#5-464	93-3381					
C\$AUTO	=	000061	#5-464	47-1749	49-1837	52-1960	62-2284	82-2911	84-3001
C\$BRK	=	000022	89-3167	89-3169					84-3002
									89-3164
C\$BSEG	=	000004	#5-464						
C\$BSUB	=	000002	#5-464						
C\$CEFG	=	000045	#5-464						
C\$CLCK	=	000062	#5-464						
C\$CLEA	=	000012	#5-464	95-3430					
C\$CLOS	=	000035	#5-464						
C\$CLP1	=	000006	#5-464						
C\$CVEC	=	000036	#5-464	82-2967	82-2969	93-3380			
C\$DCLN	=	000044	#5-464	49-1839	91-3263	91-3283			
C\$DODU	=	000051	#5-464	72-2746	93-3392				

CZTUJDO CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 3
CREF

SEQ 0130

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

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07 08

PAGE 4
CREF

SEQ 0131

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
DEV5		004652	19-992 #19-1004
DEV6		005064	19-993 #19-1005
DEV7		005276	19-994 #19-1006
DFPTBL	C	002176	#8-552
DFTL1		012774	*72-2712 *72-2713 #72-2714 72-2714
DIAGMC	=	000000	5-464 5-464
DLV	=	000074 G	#16-924 56-2047 58-2133 58-2164 *62-2265 62-2266 62-2267 *62-2269 *62-2290
			62-2291 62-2292 *62-2294 66-2415 74-2770 74-2772 *74-2773 *84-2987 84-2988
			84-2989 *84-2991 *91-3311
DOBRK	014010 G		49-1800 49-1820 #82-2905 117-3851
DONE	003322 G		#15-863 *45-1725 47-1745 *102-3583 *104-3605 *106-3647 *107-3682 *109-3720 *111-3751
			*113-3782 *115-3813 *117-3912
DR	=	000060 G	#16-917 37-1628 37-1630 *37-1632 *39-1660 39-1661 *39-1663 74-2766 76-2797
			*91-3274 91-3278 91-3280 104-3600 106-3631 106-3631 106-3632 107-3666
			107-3666 107-3667 107-3667 109-3708 109-3709 109-3709 111-3739 111-3740 111-3740
			113-3770 113-3771 113-3771 115-3801 115-38 115-3802
DRVCHK	002220		#9-580 109-3706 111-3737 113-3768 115-3749
EF.CON	=	000036 G	#10-653
EF.NEW	=	000035 G	#10-653
EF.PWR	=	000034 G	#10-653
EF.RES	=	000037 G	#10-653
EF.STA	=	000040 G	#10-653 91-3242
ENDGP8	010146		58-2089 58-2186 #58-2188
ENDRSP	007100		50-1888 #50-1940
ENDT9	027206		117-3916 117-3931 #117-3936
ERRDES	013170 G		72-2700 72-2714 72-2720 72-2727 72-2738 72-2742 #74-2759 86-3051
ERRMOD	010150		58-2153 #58-2190
ESABO	=	177720 G	#12-747
ESCKS	=	177757 G	#12-758 12-760 12-761 12-762 70-2615
ESCKSM	=	177757 G	#12-760
ESCMD	=	177720 G	#12-756 70-2639
ESNCRT	=	177767 G	#12-748 70-2634
ESNOMO	=	177737 G	#12-755 70-2610
ESNONX	=	177770 G	#12-749 70-2644
ESOK	=	000000 G	#12-750 70-2598
ESPART	=	177776 G	#12-751 70-2654
ESRD	=	177757 G	#12-762
ESREC	=	177711 G	#12-757 70-2659
ESSK	=	177740 G	#12-752 70-2628
ESSLF	=	177777 G	#12-759
ESTRY	=	000001 G	#12-753 70-2601
ESWLOC	=	177765 G	#12-754 70-2649
ESWR	=	177757 G	#12-761
EVL	=	000004 G	#10-653 72-2708
EVLTHR	002222		#9-581 72-2710
EXOFF	007427		#56-2062 58-2177 62-2277 62-2285 62-2297
EXON	007426		56-2018 #56-2061 58-2106
ESEND	=	002100	#5-464 5-507
ESLOAD	=	000035	#5-464
FLGLOC	016752 G		72-2708 *91-3320 #91-3359
FM	015730		89-3185 89-3195 #89-3218
FMO	015712		89-3184 #89-3215

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 5
CREF

SEQ 0132

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
FTLNM		003320	#15-862 72-2731
FSAU	=	000015	#5-464 99-3490 99-3513
FSAUTO	=	000020	#5-464 93-3367 93-3381
F\$BGN	=	000040	#5-464 5-490 6-516 9-593 10-646 74-2759 84-2976 84-2984 88-3105
			89-3149 89-3156 89-3224 91-3234 93-3367 95-3404 97-3440 99-3490 102-3575
			102-3578 102-3580 102-3587 104-3592 104-3594 104-3614 106-3622 106-3624 106-3649
			107-3657 107-3659 107-3691 109-3697 109-3699 109-3723 111-3728 111-3730 111-3754
			113-3759 113-3761 113-3785 115-3790 115-3792 115-3816 117-3821 117-3828 117-3938
			121-3989 123-4054 123-4065 125-4108 127-4143 127-4145 127-4146 127-4146 127-4151
			127-4152
F\$CLEA	=	000007	#5-464 95-3404 95-3430
F\$DU	=	000016	#5-464 97-3440 97-3468
F\$END	=	000041	#5-464 5-464 5-464 5-464 5-464 5-464 5-464 5-464 5-464 5-464
			5-464 5-464 5-464 5-464 5-464 5-464 5-464 5-464 5-464 5-464
			9-593 10-646 74-2776 84-2980 84-2993 88-3105 89-3149 89-3208 89-3224
			91-3350 93-3381 95-3430 97-3468 99-3513 102-3575 102-3578 102-3578 102-3578
			102-3580 102-3587 102-3587 104-3592 104-3592 104-3592 104-3594 104-3614 104-3614
			106-3622 106-3622 106-3622 106-3624 106-3649 106-3649 107-3657 107-3657 107-3657
			107-3659 107-3691 107-3691 109-3697 109-3697 109-3697 109-3699 109-3723 109-3723
			111-3728 111-3728 111-3728 111-3730 111-3754 111-3754 113-3759 113-3759 113-3759
			113-3761 113-3785 113-3785 115-3790 115-3790 115-3790 115-3792 115-3816 115-3816
			117-3821 117-3821 117-3821 117-3828 117-3938 117-3938 121-3989 123-4054 123-4080
			125-4122 127-4143 127-4145 127-4146 127-4151 127-4152
F\$HARD	=	000004	#5-464 123-4065 123-4080
F\$HW	=	000013	#5-464 8-552 8-565
F\$INIT	=	000006	#5-464 91-3234 91-3350
F\$JMP	=	000050	#5-464 102-3580 104-3594 106-3624 107-3659 109-3699 111-3730 113-3761 115-3792
			117-3828
F\$MOD	=	000000	#5-464 5-490 9-593 10-646 88-3105 89-3149 89-3224 102-3575 121-3989
			123-4054 127-4143
F\$MSG	=	000011	#5-464 74-2759 74-2776
F\$PROT	=	000021	#5-464 6-516 6-520
F\$PWR	=	000017	#5-464
F\$RPT	=	000012	#5-464 89-3156 89-3208
F\$SEG	=	000003	#5-464
F\$SOFT	=	000005	#5-464 125-4108 125-4122
F\$SRV	=	000010	#5-464 84-2976 84-2980 84-2984 84-2993
F\$SUB	=	000002	#5-464
F\$SW	=	000014	#5-464 9-574 9-591
F\$TEST	=	000001	#5-464 102-3578 102-3587 104-3592 104-3614 106-3622 106-3649 107-3657 107-3691
			109-3697 109-3723 111-3728 111-3754 113-3759 113-3785 115-3790 115-3816 117-3821
			117-3938
GBTMP	010572		*62-2261 *62-2271 *62-2281 62-2282 *62-2301 #62-2307
GBTMP2	010574		*62-2276 62-2287 62-2289 *62-2296 62-2299 #62-2308
GETANS	007152	G	47-1747 #54-1984
GETHRD	016316		91-3261 #91-3265
GETPTR	007216		#54-1976
GETR5	017222		97-3444 #97-3469
GTAGIN	007464		#58-2091 58-2183
GTBYTE	010346	G	56-2026 56-2046 58-2110 58-2130 58-2161 #62-2261
GTDOWN	010104		58-2095 58-2099 58-2114 58-2125 58-2129 58-2132 58-2136 58-2163 58-2167
			58-2169 58-2175 #58-2179

CZIUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 6
CREF

SEQ 0133

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
GTOK		007730	58-2118 #58-2138
GTPKS1	G	007220	54-1993 #56-2013
GTPKS8	G	007430	54-1989 #58-2084
GTPTR		010344	*58-2090 58-2091 58-2187 58-2180 *58-2182 #60-2235
GTUM		007672	58-2123 #58-2127
GSCNT0	=	000200	#5-464
GSDLM	=	000372	#5-464
GSDISP	=	000003	#5-464
GSEXCP	=	000400	#5-464
GSHILI	=	000002	#5-464
GSLOLI	=	000001	#5-464
GSNO	=	000000	#5-464
GSOFFS	=	000400	#5-464 123-4068 123-4069 123-4070 123-4071 123-4072 125-4109 125-4110 125-4111
			125-4112 125-4113 125-4114 125-4115
GSOFSI	=	000376	#5-464 123-4068 123-4069 123-4070 123-4071 123-4072 125-4109 125-4110 125-4111
			125-4112 125-4113 125-4114 125-4115
GSPRMA	=	000001	#5-464
GSPRMD	=	000002	#5-464
GSPRML	=	000000	#5-464 123-4068 123-4069 123-4070 123-4071 123-4072 125-4110 125-4111 125-4112 125-4113 125-4115
GSRADA	=	000140	#5-464
GSRADB	=	000000	#5-464
GSRADD	=	000040	#5-464 125-4109 125-4114
GSRADL	=	000120	#5-464 123-4070 123-4071 123-4072 125-4110 125-4111 125-4112 125-4113 125-4115
GSRADO	=	000020	#5-464 123-4068 123-4069
GSXFER	=	000004	#5-464
GSYES	=	000010	#5-464 123-4068 123-4069 123-4070 123-4071 123-4072 125-4109 125-4110 125-4111
			125-4112 125-4113 125-4114 125-4115
HARDR	=	000136	G #17-954 89-3180 89-3190
HARDW	=	000140	G #17-955 89-3182 89-3192
HELP	=	000000	#5-449 5-459 5-481 5-499 7-522 7-537 8-559 9-584 #10-598
			10-636 10-655 19-1007 19-1013 21-1029 21-1034 21-1042 21-1049 21-1054
			21-1060 37-1556 37-1568 37-1573 37-1579 37-1584 37-1590 37-1598 37-1605
			37-1611 37-1617 #89-3111 91-3328 91-3338 95-3411 95-3418 97-3450 97-3456
			99-3493 99-3499 #100-3519 100-3560 100-3566 121-3990 121-3995 121-4005 #123-4016
			123-4074 123-4090 125-4116 127-4136
HOE	=	100000	G #10-653
IRD	=	013126	72-2732 #72-2740
HRDRD	=	000016	G #11-679 68-2554
HRDWR	=	000020	G #11-680 68-2556
HRD1	=	011746	68-2541 #68-2550
IBE	=	010000	G #10-653
IDPTR	=	003324	G #15-864 *45-1726 45-1727 45-1729 *45-1731
IDU	=	000040	G #10-653
IER	=	020000	G #10-653
INIT		016164	#91-3236
INITWD		014542	*82-2905 82-2932 82-2940 82-2947 82-2954 82-2960 #84-3010
INIT2		016212	91-3243 #91-3245
ISR	=	000100	G #10-653
IXE	=	004000	G #10-653
ISAU	=	000041	#5-464 #99-3490 #99-3513
ISAUTO	=	000041	#5-464 #93-3367 #93-3381
ISCLN	=	000041	#5-464 #95-3404 #95-3430

CZTUUDO CRFATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 7
CREF

SEQ 0134

SYMBOL	VALUE	CROSS REFERENCE	REFERENCES
ISDU	= 000041		#5-464 #97-3440 #97-3468
ISHRD	= 000041		#123-4065 #123-4080
ISINIT	= 000041		#5-464 #91-3234 #91-3350
ISMOD	= 000041		#5-464 5-490 #5-490 9-593 #9-593 10-646 #10-646 88-3105 #88-3105
			89-3149 #89-3149 89-3224 #89-3224 102-3575 #102-3575 121-3989 #121-3989 123-4054
			#123-4054 127-4143 #127-4143
ISMSG	= 000041		#5-464 #74-2759 #74-2776
ISPROT	= 000040		#5-464 #6-516
ISPTAB	= 000041		#5-464 127-4146 #127-4146 127-4151 #127-4151
ISPR	= 000041		#5-464
ISRPT	= 000041		#5-464 #89-3156 #89-3208
ISSEG	= 000041		#5-464 102-3578 104-3592 106-3622 107-3657 109-3697 111-3728 113-3759 115-3790
			117-3821
ISSETU	= 000041		#5-464 127-4145 #127-4145 127-4146 127-4152 #127-4152
ISSFT	= 000041		#125-4108 #125-4122
ISSRV	= 000041		#5-464 #84-2976 #84-2980 #84-2984 #84-2993
ISSUB	= 000041		#5-464 102-3578 104-3592 106-3622 107-3657 109-3697 111-3728 113-3759 115-3790
			117-3821
ISTST	= 000041		#5-464 102-3578 #102-3578 102-3580 102-3587 #102-3587 #102-3587 104-3592 #104-3592
			104-3594 104-3614 #104-3614 #104-3614 106-3622 #106-3622 106-3624 106-3649 #106-3649
			#106-3649 107-3657 #107-3657 107-3659 107-3691 #107-3691 #107-3691 109-3697 #109-3697
			109-3699 109-3723 #109-3723 #109-3723 111-3728 #111-3728 111-3730 111-3754 #111-3754
			#111-3754 113-3759 #113-3759 113-3761 113-3785 #113-3785 #113-3785 115-3790 #115-3790
			115-3792 115-3816 #115-3816 115-3816 117-3821 #117-3821 117-3828 117-3938 #117-3938
			#117-3938
JSJMP	= 000167		#5-464
LENGTH	002210		#9-576 91-3322
LGOFST	= 000120	G	#17-944 72-2692
LNCNT	015130		*88-3084 *88-3089 #88-3098
LOE	= 040000	G	#10-653
LOG	012634	G	52-1966 62-2305 66-2398 66-2418 66-2427 66-2455 68-2505 68-2528 68-2557
			70-2665 #72-2684 82-2918 82-2962 84-3006
LOGO	013156		72-2721 72-2739 #72-2747
LOGOK	012722		72-2699 #72-2702
LOGOK2	013006		72-2709 72-2711 #72-2716
LOGU	012710		#72-2700
LOG1	013024		*72-2718 *72-2719 #72-2720 72-2720
LOG2	013054		*72-2725 *72-2726 #72-2727 72-2727
LOG3	013114		*72-2736 *72-2737 #72-2738 72-2738
LOG38	013136		*72-2740 *72-2741 #72-2742 72-2742
LOT	= 000010	G	#10-653
LSTDEV	003366	G	#19-994 37-1634 39-1664 41-1686 45-1729 49-1807 49-1830 58-2180 64-2341
			89-3196 91-3254 93-3376 117-3930
LSACP	002110	G	#5-507
LSAPT	002036	G	#5-507
LSAU	017302	G	5-507 #99-3490
LSAUT	002070	G	#5-507
LSAUTO	016754	G	5-507 #93-3367
LSCCP	002106	G	#5-507
LSCLEA	017136	G	5-507 #95-3404
LSCO	002032	G	#5-507
LSDEPO	002011	G	#5-507

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 8
CRCF

SEQ 0135

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
LSDESC	002122 G	5-507	#5-509
LSDESP	002076 G	#5-507	
LSDEVP	002060 G	#5-507	
LSDISP	002152 G	5-507	#7-535
LSDLY	002116 G	#5-507	
LSDTP	002040 G	#5-507	
LSDTYP	002034 G	#5-507	
LSDU	017156 G	5-507	#97-3440
LSDUT	002072 G	#5-507	
LSDVTY	005510 G	5-507	#21-1027
LSEF	002052 G	#5-507	
LSENV1	002044 G	#5-507	
LSETP	002102 G	#5-507	
LSEXP1	002046 G	#5-507	
LSEXP4	002064 G	#5-507	
LSEXP5	002066 G	#5-507	
LSHARD	041342 G	5-507	123-4065 #123-4065
LSHIME	002120 G	#5-507	
LSHPCP	002016 G	#5-507	
LSHPTP	002022 G	#5-507	
LSHW	002176 G	5-507	8-552 #8-552
LSICP	002104 G	#5-507	
LSINIT	016164 G	5-507	#91-3234
LSLADP	002026 G	#5-507	
LSLAST	042150 G	5-507	#127-4142 127-4152
LSLOAD	002100 G	#5-507	
LSLUN	002074 G	#5-507	*72-2689 *72-2690 *86-3049 *86-3050 *91-3268
LSMREV	002050 G	#5-507	
LSNAME	002000 G	#5-507	
LSPRIO	002042 G	#5-507	
LSPROT	002142 G	5-507	#6-516
LSPRT	002112 G	#5-507	
LSREPP	002062 G	#5-507	
LSREV	002010 G	#5-507	
LSRPT	015150 G	5-507	#89-3156
LSOFT	041504 G	5-507	125-4108 #125-4108
LSGPC	002056 G	#5-507	
LSGPCP	002020 G	#5-507	
LSPTP	002024 G	#5-507	
LSSTA	002030 G	#5-507	
LSW	002210 G	5-507	9-574 #9-574
LSTEST	002114 G	#5-507	
LSIML	002014 G	#5-507	
LSUNIT	002012 G	#5-507	91-3260 91-3316
L10001	002206	8-552	#8-565
L10002	002226	9-574	#9-591
L10003	013360	#74-2776	
L10004	014446	#84-2980	
L10005	014502	#84-2993	
L10006	015572	#87-3208	
L10007	016662	#91-3350	
L10010	017060	#93-3381	

CZTUUD0 CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 9
CREF

SEQ 0136

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

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 10
CREF

SEQ 0137

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

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 11
CREF

SEQ 0138

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

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 12

SEQ 0139

SYMBOL CROSS REFERENCE

CREF

SYMBOL	VALUE	REFERENCES
RSMSIZ	= 000012 G	#12-722 12-730 102-3582 102-3582 104-3600 104-3600 106-3631 106-3631 106-3631
		106-3631 106-3632 106-3632 106-3632 107-3666 107-3666 107-3666 107-3666 107-3666
		107-3667 107-3667 107-3667 107-3667 109-3709 109-3709 109-3709 109-3709 111-3740
		111-3740 111-3740 111-3740 113-3771 113-3771 113-3771 113-3771 115-3802 115-3802
		115-3802 115-3802 117-3833 117-3840 117-3860 117-3868 117-3887 117-3895 117-3895
RSNDSZ	= 000016 G	#12-720 12-726 12-731 56-2035 58-2122 102-3582 104-3600 106-3631 106-3631
		106-3632 106-3632 107-3666 107-3666 107-3667 107-3667 109-3709 109-3709 111-3740
		111-3740 113-3771 113-3771 115-3802 115-3802 117-3876 117-3903 117-3903
RSNTAB	002226	#13-770 72-2705
RSSEND	= 000100 G	#12-736
RSSGET	= 000012 G	#12-740 117-3834
RSSNIT	= 000001 G	#12-742 82-2908 117-3861
RSSNOP	= 000000 G	#12-741 117-3888
RSSNSZ	= 000016 G	#12-730 66-2443 102-3582 104-3600 106-3631 106-3631 106-3632 106-3632 107-3666
		107-3666 107-3667 107-3667 109-3709 109-3709 111-3740 111-3740 113-3771 113-3771
		115-3802 115-3802 117-3842 117-3874 117-3901 117-3901 117-3901 117-3901 117-3901
RSSRD	= 000002 G	#12-738 50-1916 70-2603 106-3631 106-3632 106-3632 107-3666 107-3667 107-3667
		109-3709 111-3740 111-3740 113-3771 115-3802 115-3802 115-3802 115-3802 115-3802
RSSSEK	= 000005 G	#12-739 104-3600
RSSSLF	= 000007 G	#12-743 70-2620 102-3582
RSSWR	= 000003 G	#12-737 50-1925 70-2607 106-3631 107-3666 109-3709 113-3771 113-3771
RSVP	006560 G	#50-1880 102-3582 104-3600 106-3631 106-3631 106-3631 106-3632 106-3632 107-3666
		107-3666 107-3666 107-3667 107-3667 109-3709 109-3709 109-3709 111-3740 111-3740
		113-3771 113-3771 113-3771 115-3802 115-3802 117-3849 117-3879 117-3906
RSXOFF	= 000023 G	#12-714 56-2062 121-3971 121-3972
RSXON	= 000020 G	#12-713 56-2061
RTRYN	012204	68-2512 68-2543 #68-2574
RUN	006052 G	#47-1743 47-1752 102-3579 102-3579 104-3593 104-3593 104-3593 106-3623 106-3623 107-3658
		107-3658 109-3698 109-3698 111-3729 111-3729 113-3760 113-3760 115-3791 115-3791
		117-3825 117-3825
SAVCNT	= 000206 G	#17-975 49-1787 *49-1798
SECREC	003334 G	#15-868 *91-3324 *91-3327 109-3717 111-3748 113-3779 115-3810
SERVST	010342	*54-1987 58-2184 *60-2220 #60-2234
SETDR	005632 G	#39-1658 102-3579 104-3593 106-3623 107-3658 109-3628 111-3729 113-3760 115-3791
SETLEN	016624	#91-3322
SETPTR	005706	*39-1658 39-1659 39-1664 *39-1666 #39-1669
SETSRV	010256	58-2094 58-2098 #60-2210
SETUP	006004 G	#45-1725 102-3579 102-3579 104-3593 104-3593 106-3623 106-3623 107-3658 107-3658
		109-3698 109-3698 111-3729 111-3729 113-3760 113-3760 115-3791 115-3791 117-3824
SFPTBL	002210 G	#9-574
SFT	013104	72-2733 #72-2736
SFTOUT	041566	#125-4122
SFTRD	= 000002 G	#11-673 68-2525
SFTWR	= 000004 G	#11-674 68-2527
SKERR	= 000024 G	#11-682 70-2630
SLFER	= 000044 G	#11-690 70-2625
SND	006650	50-1890 #50-1893 50-1897
SNDBYT	007102 G	50-1893 #52-1955 56-2019 58-2108 58-2156 58-2178 62-2278 62-2286 62-2298
SNDCNT	= 000070 G	#16-922 *50-1892 *50-1896 *102-3582 *104-3600 *106-3631 *106-3631 *106-3631
		*106-3632 *106-3632 *107-3666 *107-3666 *107-3666 *107-3666 *107-3667 *107-3667 *107-3667
		*109-3709 *109-3709 *109-3709 *111-3740 *111-3740 *111-3740 *113-3771 *113-3771 *113-3771
		*115-3802 *115-3802 *117-3842 *117-3874 *117-3901

[illegible]

REFERENCES

[illegible]

CZTUUDO		CREATED BY		MACRO ON 25-FEB-82 AT 07:08		PAGE 16		N 11		SEQ 0143	
SYMBOL CROSS REFERENCE		SYMBOL VALUE		REFERENCES		CREF					
				127-4142	127-4142	127-4142	127-4142	127-4142	127-4146	127-4146	127-4146
				127-4146	127-4146						
SVCSUB	=	000001		#5-464	#5-472						
SVCTAG	=	000001		#5-464	#5-474	8-565	9-591	74-2776	84-2980	84-2993	89-3208
				93-3381	95-3430	97-3468	99-3513	102-3587	104-3614	106-3649	107-3691
				111-3754	113-3785	115-3816	117-3938	123-4080	125-4122	127-4146	127-4151
SVCTST	=	000001		#5-464	#5-471	102-3578	104-3592	106-3622	107-3657	109-3697	111-3728
				115-3790	117-3821						
SWAPDR	005530	G		#37-1623	102-3579	104-3593	106-3623	107-3658	109-3698	111-3729	113-3760
SWPTR	005630			*37-1624	37-1625	37-1634	*37-1636	#37-1644			
SYSTAT	003306	G		#15-849	*49-1789	*49-1794	*49-1803	54-1985	*56-2030	*58-2116	64-2326
				68-2490	68-2492	*68-2561	74-2766	*78-2818	*78-2821	78-2832	*82-2960
SLSYM	=	010000		#5-464	#8-565	#9-591	#74-2776	#84-2980	#84-2993	#89-3208	#91-3350
				#95-3430	#97-3468	#99-3513	#102-3587	#104-3614	#106-3649	#107-3691	#109-3723
				#113-3785	#115-3816	#117-3938	#123-4080	#125-4122			
TAPLEN	003310	G		#15-858	*91-3322	*91-3323	91-3325	109-3703	109-3718	111-3734	111-3749
				113-3780	115-3796	115-3811					
TEST9	003342	G		#15-871	50-1885	58-2085	58-2104	58-2141	58-2172	64-2336	66-2407
				*91-3241	*117-3830	*117-3856	*117-3913				
THRSHI	012152			68-2522	#68-2572						
THRSLO	012124			68-2520	#68-2570						
TMP	=	000066	G	#16-921	*106-3627	106-3643	*106-3644	*107-3662	107-3678	*107-3679	*109-3703
				*109-3718	*111-3734	*111-3741	*111-3749	*113-3765	*113-3772	*113-3780	*115-3796
				*115-3811							*109-3710
TOMANY	016664			91-3262	#91-3353						
TORCVB	=	000050	G	#11-692	62-2304	84-3005					
TOSNDB	=	000056	G	#11-694	52-1965	82-2917					
TRBUF	027722			50-1889	50-1891	50-1898	102-3582	104-3600	106-3631	106-3631	106-3631
				106-3632	106-3632	107-3666	107-3666	107-3666	107-3667	107-3667	109-3709
				109-3709	109-3709	109-3709	111-3740	111-3740	113-3771	113-3771	113-3771
				115-3802	115-3802	117-3831	117-3858	117-3885	#121-3974		
TRK	=	000062	G	#16-918	*109-3704	109-3714	*109-3716	*111-3735	111-3745	*111-3747	*113-3766
				*113-3778	*115-3797	115-3807	*115-3809				113-3776
TRPHND	017064			93-3369	#93-3389						
TRPPTR	017062			*93-3370	93-3371	93-3376	*93-3378	#93-3382			
TSTPC	=	000020	G	#16-907	*45-1728	49-1805	49-1826	*50-1881			
TSTTOP	003326			#15-865	45-1728	*102-3579	*104-3593	*106-3623	*107-3658	*109-3698	*111-3729
				*115-3791	*117-3823						*113-3760
TST1	017350			102-3579	#102-3582						
TST2	017552			104-3593	#104-3597						
TST3	020024			106-3623	#106-3627						
TST3PT	022742			106-3630	106-3634	107-3665	107-3669	#107-3684			
TST4	021416			107-3658	#107-3662						
TST5	023022			109-3698	#109-3702						
TST5EX	023736			109-3715	#109-3720						
TST6	024012			111-3729	#111-3733						
TST6EX	024522			111-3746	#111-3751						
TST7	024576			113-3760	#113-3764						
TST7EX	025512			113-3777	#113-3782						
TST8	025566			115-3791	#115-3795						
TST8EX	026276			115-3808	#115-3813						
TST9	026330			117-3823	#117-3830						

CZTUUDO
SYMBOL CROSS REFERENCE

MACRO ON 25-FEB-82 AT 07:08

PAGE 17
CREF

B 12

SEQ 0144

SYMBOL VALUE
TUVECT = 000204 G

TSARGC = 000002

TS CODE = 006130

TSERRN = 000146

TSEXCP = 000000

TSFLAG = 000040

TSFREE = 042164

TSGMAN = 000000

TSHILI = 000376

TSLAST = 000001

TSLOLI = 000001

TSLSYM = 010000

TSLTNO = 000011

TSNEST = 177777

REFERENCES

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

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08
 SYMBOL CROSS REFERENCE
 SYMBOL VALUE

PAGE 18
 CREF

SEQ 0145

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

CZTUUDO		CREATED BY		MACRO ON 25-FEB-82 AT 07:08		PAGE 19				SEQ 0146	
SYMBOL CROSS REFERENCE				CREF							
SYMBOL	VALUE	REFERENCES									
		#117-3828	117-3828	#117-3938	117-3938	#121-3989	121-3989	#123-4068	123-4068	#123-4068	
		123-4068	#123-4068	123-4068	#123-4069	123-4069	#123-4069	123-4069	#123-4069	123-4069	
		#123-4070	123-4070	#123-4070	123-4070	#123-4070	123-4070	#123-4071	123-4071	#123-4071	
		123-4071	#123-4071	123-4071	#123-4072	123-4072	#123-4072	123-4072	#123-4072	123-4072	
		#123-4080	123-4080	#125-4109	125-4109	#125-4109	125-4109	#125-4109	125-4109	#125-4110	
		125-4110	#125-4110	125-4110	#125-4110	125-4110	#125-4111	125-4111	#125-4111	125-4111	
		#125-4111	125-4111	#125-4112	125-4112	#125-4112	125-4112	#125-4112	125-4112	#125-4113	
		125-4113	#125-4113	125-4113	#125-4113	125-4113	#125-4114	125-4114	#125-4114	125-4114	
		#125-4114	125-4114	#125-4115	125-4115	#125-4115	125-4115	#125-4115	125-4115	#125-4122	
		125-4122	#127-4143	127-4143							
T\$TEST	= 000011	#5-464	102-3578	#102-3578	102-3578	104-3592	#104-3592	104-3592	104-3592	106-3622	#106-3622
		106-3622	107-3657	#107-3657	107-3657	109-3697	#109-3697	109-3697	109-3697	111-3728	#111-3728
		111-3728	113-3759	#113-3759	113-3759	115-3790	#115-3790	115-3790	115-3790	117-3821	#117-3821
		117-3821	127-4142								
T\$TSTM	= 177777	#5-464	47-1749	49-1836	49-1837	49-1839	52-1960	58-2191	62-2284	68-2512	
		68-2517	68-2520	68-2522	68-2536	68-2543	68-2551	72-2700	72-2714	72-2720	
		72-2727	72-2738	72-2742	72-2746	74-2766	74-2769	74-2772	74-2776	82-2911	
		82-2926	82-2927	82-2929	82-2967	82-2969	84-3001	84-3002	86-3051	86-3052	
		88-3086	88-3091	88-3093	89-3164	89-3166	89-3167	89-3168	89-3169	89-3184	
		89-3185	89-3195	89-3208	91-3242	91-3262	91-3263	91-3269	91-3282	91-3283	
		91-3320	91-3350	93-3369	93-3380	93-3381	93-3389	93-3392	95-3409	95-3430	
		97-3448	97-3468	99-3513	102-3580	102-3587	104-3594	104-3614	106-3624	106-3649	
		107-3659	107-3691	109-3699	109-3723	111-3730	111-3754	113-3761	113-3785	115-3792	
		115-3816	117-3828	117-3918	117-3925	117-3927	117-3929	117-3938			
T\$TSTS	= 000001	#5-464	#102-3578	#104-3592	#106-3622	#107-3657	#109-3697	#111-3728	#113-3759	#115-3790	
		#117-3821									
T\$SAU	= 010013	#99-3490	99-3513								
T\$SAUT	= 010010	#93-3367	93-3381								
T\$SCLE	= 010011	#95-3404	95-3430								
T\$SDAT	= 010031	#127-4146	127-4146	127-4151							
T\$SDU	= 010012	#97-3440	97-3468								
T\$SHAR	= 010025	#123-4065	123-4065	123-4080							
T\$SHW	= 010001	#8-552	8-552	8-565							
T\$SINI	= 010007	#91-3234	91-3350								
T\$MSG	= 010003	#74-2759	74-2776								
T\$SPC	= 000001	#127-4145	127-4152								
T\$SPRO	= 010000	#6-516									
T\$SPTA	= 010030	#127-4145	127-4146	#127-4146							
T\$SRPT	= 010006	#89-3156	89-3208								
T\$SOF	= 010026	#125-4108	125-4108	125-4122							
T\$SRV	= 010005	#84-2976	84-2980	#84-2984	84-2993						
T\$SW	= 010002	#9-574	9-574	9-591							
T\$TES	= 010024	#102-3578	102-3580	102-3587	#104-3592	104-3594	104-3614	#106-3622	106-3624	106-3649	
		#107-3657	107-3659	107-3691	#109-3697	109-3699	109-3723	#111-3728	111-3730	111-3754	
		#113-3759	113-3761	113-3785	#115-3790	115-3792	115-3816	#117-3821	117-3828	117-3938	
T1	017304	G	7-535	#102-3578							
T1TRY	= 000146	G	#17-958								
T2	017506	G	7-535	#104-3592							
T3	017760	G	7-535	#106-3622							
T4	021352	G	7-535	#107-3657							
T4TRY	= 000132	G	#17-952								
T5	022756	G	7-535	#109-3697							

CZTUJDO CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 20
CREF

SEQ 0147

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
T6		023746 G	7-535 #111-3728
T7		024532 G	7-535 #113-3759
T8		025522 G	7-535 #115-3790
T9		026306 G	7-535 #117-3821
11AM		= 000200 G	#10-653
UNIT		013362 G	74-2766 #74-2777
UNITNO		027366	58-2191 *117-3917 117-3925 117-3927 117-3929 *117-3933 #117-3945
UNREC		012224	68-2551 #68-2576
UNSUC		011622	68-2516 #68-2533
UNXPCT		007634	#58-2119
WAIT		014504	82-2935 82-2943 82-2950 #84-2997 84-3004
WHCHDR		013640 G	50-1919 50-1933 72-2695 #76-2795
WRLOCK		= 000026 G	#11-683 70-2651 72-2723
WRTNO		= 000110 G	#16-931 *50-1935 89-3185 91-3289 91-3291
WRTN1		= 000112 G	#16-932 *50-1938 89-3195
XFNSND		005640	50-1886 #50-1891
XMDB		= 000030 G	#16-911 52-1968 82-2920 84-2979 *91-3306
XMSR		= 000026 G	#16-910 52-1957 82-2907 82-2913 82-2924 82-2934 82-2942 82-2965 84-2978
			*91-3301
XSCNT		= 000036 G	#16-914 *102-3582 *104-3600 *106-3631 *106-3631 *106-3631 *106-3631 *107-3666 *107-3666 *107-3666
			*109-3709 *109-3709 *109-3709 *113-3771 *113-3771 *113-3771 *117-3846 *117-3876 *117-3903
XSFLG		= 000034 G	#16-913 50-1904 56-2014 66-2380 *102-3582 *104-3600 *106-3631 *106-3631 *106-3631
			106-3631 106-3632 106-3632 *107-3666 *107-3666 *107-3666 107-3666 107-3667 107-3667
			*109-3709 *109-3709 *109-3709 109-3709 111-3740 111-3740 *113-3771 *113-3771 *113-3771
			113-3771 115-3802 115-3802 *117-3845 *117-3875 *117-3902
XSPKMM		= 000032 G	#16-912 50-1900 *50-1900 56-2022 58-2096 *58-2100 *58-2113 *58-2119 *58-2135
			*58-2166 66-2379 *102-3582 *104-3600 *106-3631 *106-3631 *106-3631 *106-3632
			*106-3632 *107-3666 *107-3666 *107-3666 *107-3666 *107-3667 *107-3667 *109-3709 *109-3709
			*109-3709 *109-3709 *111-3740 *111-3740 *113-3771 *113-3771 *113-3771 *113-3771 *115-3802
			*115-3802 *117-3847 *117-3877 *117-3904
XSPTR		= 000106 G	#16-930 *50-1906 58-2101 *58-2102 58-2103 *58-2179
XSALWA		= 000000	#5-464
XSALS		= 000040	#5-464
X\$OFFS		= 000400	#5-464
X\$TRUE		= 000020	#5-464

CZTUUD0 CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 21
CREF

SEQ 0148

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

PAGE
CREF

MACRO NAME	REFERENCES	#123-4070	#123-4071	#123-4072	#125-4110	#125-4111	#125-4112	#125-4113	#125-4115	
GPRML	5-507									
HEADER	127-4142									
LASTAD	#5-507	#5-507	#5-507	#5-507	#5-507					
MSBYTE	#102-3580	102-3580	#104-3594	104-3594	#106-3624	106-3624	#107-3659	107-3659	#109-3699	109-3699
MSCHEC	#111-3730	111-3730	#113-3761	113-3761	#115-3792	115-3792	#117-3828	117-3828		
MSCNTO	#123-4068	123-4068	#123-4069	123-4069	#123-4070	123-4070	#123-4071	123-4071	#123-4072	123-4072
	#125-4109	125-4109	#125-4110	125-4110	#125-4111	125-4111	#125-4112	125-4112	#125-4113	125-4113
	#125-4114	125-4114	#125-4115	125-4115						
MSCOUN	#58-2191	58-2191	#68-2512	68-2512	#68-2517	68-2517	#68-2520	68-2520	#68-2522	68-2522
	#68-2536	68-2536	#68-2543	68-2543	#68-2551	68-2551	#74-2766	74-2766	74-2766	74-2766
	#74-2769	74-2769	74-2769	74-2769	74-2769	#74-2772	74-2772	#86-3052	86-3052	#88-3086
	88-3086	#88-3091	88-3091	#88-3093	88-3093	#89-3166	89-3166	#89-3168	89-3168	#89-3184
	89-3184	#89-3185	89-3185	89-3185	89-3185	89-3185	89-3185	89-3185	89-3185	89-3185
	#89-3195	89-3195	89-3195	89-3195	89-3195	89-3195	89-3195	89-3195	89-3195	#93-3389
	93-3389	#97-3448	97-3448	#117-3918	117-3918	#117-3925	117-3925	#117-3927	117-3927	#117-3929
	117-3929									
MSDATA	#5-507	5-507	5-507	5-507	5-507	5-507	5-507	5-507	5-507	5-507
	5-507	5-507	5-507	5-507	5-507	5-507	#5-507	5-507	5-507	5-507
	5-507	5-507	5-507	5-507	5-507	5-507	5-507	5-507	5-507	5-507
	5-507	5-507	5-507	5-507	5-507	5-507	5-507	5-507	5-507	#5-509
	5-509	#21-1027	21-1027							
MSDECR	#6-520	6-520	#8-565	8-565	#9-591	9-591	#9-593	9-593	#74-2776	74-2776
	#84-2980	84-2980	#84-2993	84-2993	#88-3105	88-3105	#89-3208	89-3208	#89-3224	89-3224
	#91-3350	91-3350	#93-3381	93-3381	#95-3430	95-3430	#97-3468	97-3468	#99-3513	99-3513
	#102-3587	102-3587	#104-3614	104-3614	#106-3649	106-3649	#107-3691	107-3691	#109-3723	109-3723
	#111-3754	111-3754	#113-3785	113-3785	#115-3816	115-3816	#117-3938	117-3938	#121-3989	121-3989
MSDEFA	#123-4080	123-4080	#125-4122	125-4122	#127-4143	127-4143	#127-4146	127-4146		
	#123-4068	#123-4068	#123-4069	#123-4069	#123-4070	#123-4070	#123-4071	#123-4071	#123-4072	#123-4072
	#125-4109	#125-4109	#125-4110	#125-4110	#125-4111	#125-4111	#125-4112	#125-4112	#125-4113	#125-4113
	#125-4114	#125-4114	#125-4115	#125-4115						
MSENDE	#8-565	#9-591	#9-593	#74-2776	#84-2980	#84-2993	#88-3105	#89-3208	#89-3224	#91-3350
	#93-3381	#95-3430	#97-3468	#99-3513	#102-3587	#104-3614	#106-3649	#107-3691	#109-3723	#111-3754
	#113-3785	#115-3816	#117-3938	#121-3989	#123-4080	#125-4122	#127-4143			
MSERRI	#49-1836	#49-1836	#72-2700	#72-2700	#72-2714	#72-2714	#72-2720	#72-2720	#72-2727	#72-2727
	#72-2738	#72-2738	#72-2742	#72-2742	#86-3051	#86-3051	#91-3262	#91-3262	#91-3282	#91-3282
MSEXCP	#123-4068	123-4068	123-4068	#123-4069	123-4069	123-4069	#125-4109	125-4109	125-4109	#125-4114
	125-4114	125-4114								
MSEXIT	#102-3580	102-3580	#104-3594	104-3594	#106-3624	106-3624	#107-3659	107-3659	#109-3699	109-3699
	#111-3730	111-3730	#113-3761	113-3761	#115-3792	115-3792	#117-3828	117-3828		
MSEXSE	#102-3580	#104-3594	#106-3624	#107-3659	#109-3699	#111-3730	#113-3761	#115-3792	#117-3828	
MSEXTJ	#102-3580	#104-3594	#106-3624	#107-3659	#109-3699	#111-3730	#113-3761	#115-3792	#117-3828	
MSGEN	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507
	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507
	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507
	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507
	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507
	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507
	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507
	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507
	#5-507	#5-507	#5-507	#5-507	#5-507	#5-507	#5-509	#5-509	#6-516	#6-516
	#7-535	#7-535	#8-552	#8-552	#8-552	#8-552	#8-565	#8-565	#9-574	#9-574
	#9-574	#9-574	#9-591	#9-591	#21-1027	#21-1027	#74-2759	#74-2759	#74-2776	#74-2776

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08
 MACRO CROSS REFERENCE
 MACRO NAME REFERENCES

PAGE 24
 CREF

SEQ 0151

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

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

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 26

SEQ 0153

MACRO CROSS REFERENCE

MACRO NAME

REFERENCES

MSPUT

#109-3697	#109-3697	#111-3728	#111-3728	#113-3759	#113-3759	#115-3790	#115-3790	#117-3821	#117-3821
#123-4054	#123-4054	#123-4065	#123-4065	#125-4108	#125-4108				
#58-2191	#58-2191	#58-2191	#58-2191	#68-2512	#68-2512	#68-2512	#68-2512	#68-2517	#68-2517
#68-2517	#68-2517	#68-2520	#68-2520	#68-2520	#68-2522	#68-2522	#68-2522	#68-2536	#68-2536
#68-2536	#68-2543	#68-2543	#68-2543	#68-2543	#68-2551	#68-2551	#68-2551	#74-2766	#74-2766
#74-2766	#74-2766	#74-2766	#74-2766	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769
#74-2769	#74-2772	#74-2772	#74-2772	#74-2772	#82-2927	#82-2927	#82-2927	#82-2927	#82-2927
#82-2929	#82-2929	#82-2929	#82-2929	#82-2929	#86-3052	#86-3052	#86-3052	#86-3052	#88-3086
#88-3086	#88-3086	#88-3086	#88-3091	#88-3091	#88-3091	#88-3093	#88-3093	#88-3093	#89-3166
#89-3166	#89-3166	#89-3168	#89-3168	#89-3168	#89-3184	#89-3184	#89-3184	#89-3184	#89-3185
#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185
#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195
#89-3195	#93-3369	#93-3369	#93-3369	#93-3369	#93-3369	#93-3389	#93-3389	#93-3389	#97-3448
#97-3448	#97-3448	#97-3448	#117-3918	#117-3918	#117-3918	#117-3925	#117-3925	#117-3925	#117-3925

MSPUT1

#117-3927	#117-3927	#117-3927	#117-3927	#117-3929	#117-3929	#117-3929	#117-3929		
#58-2191	#58-2191	#58-2191	#58-2191	#58-2191	#58-2191	#58-2191	#58-2191	#68-2512	#68-2512
#68-2512	#68-2512	#68-2517	#68-2517	#68-2517	#68-2517	#68-2517	#68-2517	#68-2520	#68-2520
#68-2520	#68-2520	#68-2522	#68-2522	#68-2522	#68-2522	#68-2522	#68-2522	#68-2536	#68-2536
#68-2543	#68-2543	#68-2543	#68-2543	#68-2543	#68-2543	#68-2543	#68-2543	#68-2551	#68-2551
#74-2766	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766	#74-2766
#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769	#74-2769
#74-2769	#74-2769	#74-2772	#74-2772	#74-2772	#74-2772	#74-2772	#74-2772	#82-2927	#82-2927
#82-2927	#82-2927	#82-2927	#82-2927	#82-2927	#82-2927	#82-2927	#82-2927	#82-2929	#82-2929
#82-2929	#82-2929	#82-2929	#82-2929	#82-2929	#86-3052	#86-3052	#86-3052	#86-3052	#86-3052
#88-3086	#88-3086	#88-3086	#88-3086	#88-3086	#88-3086	#88-3086	#88-3091	#88-3091	#88-3091
#88-3093	#88-3093	#88-3093	#88-3093	#88-3093	#89-3166	#89-3166	#89-3166	#89-3166	#89-3168
#89-3168	#89-3168	#89-3184	#89-3184	#89-3184	#89-3184	#89-3184	#89-3184	#89-3184	#89-3185
#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185
#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185	#89-3185
#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195
#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195	#89-3195
#93-3369	#93-3369	#93-3369	#93-3369	#93-3369	#93-3369	#93-3389	#93-3389	#93-3389	#93-3389
#97-3448	#97-3448	#97-3448	#97-3448	#97-3448	#97-3448	#97-3448	#117-3918	#117-3918	#117-3918
#117-3925	#117-3925	#117-3925	#117-3925	#117-3925	#117-3925	#117-3925	#117-3927	#117-3927	#117-3927
#117-3927	#117-3927	#117-3929	#117-3929	#117-3929	#117-3929	#117-3929	#117-3929	#117-3929	#117-3929

MSRADI

#123-4068	#123-4068	#123-4069	#123-4069	#123-4070	#123-4070	#123-4071	#123-4071	#123-4072	#123-4072
#125-4109	#125-4109	#125-4110	#125-4110	#125-4111	#125-4111	#125-4112	#125-4112	#125-4113	#125-4113
#125-4114	#125-4114	#125-4115	#125-4115						

MSRNRO

MSSETS

#91-3269	#91-3269	#91-3320	#91-3320						
#5-490	#5-490	#6-516	#6-516	#8-552	#8-552	#9-574	#9-574	#10-646	#10-646
#74-2759	#74-2759	#84-2976	#84-2976	#84-2984	#84-2984	#89-3149	#89-3149	#89-3156	#89-3156
#91-3234	#91-3234	#93-3367	#93-3367	#95-3404	#95-3404	#97-3440	#97-3440	#99-3490	#99-3490
#102-3575	#102-3575	#102-3578	#102-3578	#104-3592	#104-3592	#106-3622	#106-3622	#107-3657	#107-3657
#109-3697	#109-3697	#111-3728	#111-3728	#113-3759	#113-3759	#115-3790	#115-3790	#117-3821	#117-3821

MSSVC

#123-4054	#123-4054	#123-4065	#123-4065	#125-4108	#125-4108				
#47-1749	47-1749	49-1836	#49-1837	49-1837	#49-1839	49-1839	#52-1960	52-1960	#58-2191
58-2191	#62-2284	62-2284	#68-2512	68-2512	#68-2517	68-2517	#68-2520	68-2520	#68-2522
68-2522	#68-2536	68-2536	#68-2543	68-2543	#68-2551	68-2551	72-2700	72-2714	72-2720
72-2727	72-2738	72-2742	#72-2746	72-2746	#74-2766	74-2766	#74-2769	74-2769	#74-2772
74-2772	#74-2776	74-2776	#82-2911	82-2911	#82-2926	82-2926	#82-2927	82-2927	#82-2929
82-2929	#82-2967	82-2967	#82-2969	82-2969	#84-3001	84-3001	#84-3002	84-3002	86-3051
#86-3052	86-3052	#88-3086	88-3086	#88-3091	88-3091	#88-3093	38-3093	#89-3164	89-3164

MACRO CROSS REFERENCE

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

CZTUUDO CREATED BY MACRO ON 25-FEB-82 AT 07:08

PAGE 28
CREF

SEQ 0155

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