.REM_

IDENTIFICATION

PRODUCT ID: AC-T716A-MC

PRODUCT TITLE: CZTSAAO TSU05 DIAG PART 1

DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PPG

DATE: JUNE 03, 1983

TABLE OF CONTENTS

```
3237 044022                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     044022  104455                                                                TRAP    C$ERDF
     044024  002131                                                                .WORD   1113
     044026  003644'                                                              .WORD   SFIERR
     044030  011644'                                                              .WORD   SFIMSG
3238 044032                          3$:
3239 044032  012704  044530'              MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
3240 044036  004737  010472'              JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3241 044042  103404                      BCS     4$                      ;BR, IF COMMAND ISSUED OK
3245 044044                              ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     044044  104456                                                                TRAP    C$ERHRD
     044046  002132                                                                .WORD   1114
     044050  005050'                                                              .WORD   WRTMSG
     044052  011644'                                                              .WORD   SFIMSG
3246 044054                          4$:
3247 044054  004737  045334'              JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3248 044060  012704  044460'              MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3249 044064                          5$:
3250 044064  005037  002222'        10$:  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3251 044070  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
3252 044074  004737  016146'              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3253 044100  103405                      BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3254 044102  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3258 044104                              ERRDF   ERRNO,T11SR2,PKTSSR     ;DEVICE FATAL. SSR FAILED TO SET
     044104  104455                                                                TRAP    C$ERDF
     044106  002133                                                                .WORD   1115
     044110  045100'                                                              .WORD   T11SR2
     044112  011656'                                                              .WORD   PKTSSR
3259 044114                          15$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044114  104406                                                                TRAP    C$CLP1
3260 044116                              ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     044116  104410                                                                TRAP    C$ESCAPE
     044120  000074                                                                .WORD   10000$-.
3261 044122  005737  002222'              TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
3262 044126  001004                      BNE     22$                     ;BRANCH IF YES
3266 044130                              ERRHRD  ERRNO,T11NINT,PKTSSR
     044130  104456                                                                TRAP    C$ERHRD
     044132  002134                                                                .WORD   1116
     044134  045154'                                                              .WORD   T11NINT
     044136  011656'                                                              .WORD   PKTSSR
3267 044140  016501  000002        22$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
3268 044144  012702  000200              MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
3269 044150  032701  000100              BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
3270 044154  001402                      BEQ     25$                     ;BRANCH IF NOT OFF-LINE
3271 044156  052702  000100              BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
3272 044162  020201              25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
3273 044164  001404                      BEQ     30$                     ;OKAY IF MATCH
3277 044166                              ERRHRD  ERRNO,T113REJ,PKTSSR    ;COMMAND NOT ACCEPTED
     044166  104456                                                                TRAP    C$ERHRD
     044170  002135                                                                .WORD   1117
     044172  044713'                                                              .WORD   T113REJ
     044174  011656'                                                              .WORD   PKTSSR
3278 044176                          30$:
3279 044176  004737  010724'        35$:  JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
3280 044202  103405                      BCS     59$                     ;RAM OK GO ON
3284 044204                              ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     044204  104456                                                                TRAP    C$ERHRD
```

AND THE REGISTER MODIFICATION REFUSED (RMR) LOGIC.

TEST 11: NON-TAPE MOTION COMMANDS

THIS TEST VERIFIES PROPER OPERATION OF THE INITIALIZE COMMAND. TWO SUBTESTS ARE USED. THE FIRST VERIFIES THAT THE COMMAND RUNS TO COMPLETION AND STORES A VALID MESSAGE PACKET. THE SECOND VERIFIES THAT NON-ZERO VALUES IN THE COMMAND MODE FIELD CAUSES COMMAND REJECT.

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

```
                                        .TITLE  TSV2 - PROGRAM HEADER
                                        .SBTTL  PROGRAM HEADER
000000                                  .PSECT  ABS

                                        .MCALL  SVC
000000                                  SVC                     ; INITIALIZE SUPERVISOR MACROS
                                        .ENABLE LC
                                        .NLIST  BEX,CND
                                        .ENABL  AMA
        002000'                         .=.+2000
                                ;       .=2000
002000                                  BGNMOD  TSV2
002000                          TSV2::

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


002000                                  POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
002000                                  HEADER  CZTSA,A,0,655.,0
002000                          L$NAME::                ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  132                             .ASCII /Z/
002002  124                             .ASCII /T/
002003  123                             .ASCII /S/
002004  101                             .ASCII /A/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                          L$REV::                 ;REVISION LEVEL
002010  101                             .ASCII  /A/
002011                          L$DEPO::                ;0
002011  060                             .ASCII  /0/
002012                          L$UNIT::                ;NUMBER OF UNITS
002012  000000                          .WORD   0
002014                          L$TIML::                ;LONGEST TEST TIME
002014  001217                          .WORD   655.
002016                          L$HPCP::                ;POINTER TO H.W. QUES.
002016  045444'                         .WORD   L$HARD
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  045576'                         .WORD   L$SOFT
002022                          L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002154'                         .WORD   L$HW
002024                          L$SPTP::                ;PTR. TO S.W. PTABLE
002024  002164'                         .WORD   L$SW
002026                          L$LADP::                ;DIAG. END ADDRESS
002026  045672'                         .WORD   L$LAST
002030                          L$STA::                 ;RESERVED FOR APT STATS
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                          .WORD   0
```

PROGRAM HEADER

```
002036                          L$APT::                 ;APT EXPANSION
002036  000000                          .WORD   0
002040                          L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002124'                        .WORD   L$DISPATCH
002042                          L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000000                          .WORD   0
002044                          L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
002044  000000                          .WORD   0
002046                          L$EXP1::                ;EXPANSION WORD
002046  000000                          .WORD   0
002050                          L$MREV::                ;SVC REV AND EDIT #
002050     003                          .BYTE   C$REVISION
002051     003                          .BYTE   C$EDIT
002052                          L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                          .WORD   0
002054  000000                          .WORD   0
002056                          L$SPC::
002056  000000                          .WORD   0
002060                          L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  003400'                        .WORD   L$DVTYP
002062                          L$REPP::                ;PTR. TO REPORT CODE
002062  022434'                        .WORD   L$RPT
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  022122'                        .WORD   L$AU
002072                          L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022220'                        .WORD   L$DU
002074                          L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                          .WORD   0
002076                          L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  003406'                        .WORD   L$DESC
002100                          L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                        EMT     E$LOAD
002102                          L$ETP::                 ;POINTER TO ERRTBL
002102  000000                          .WORD   0
002104                          L$ICP::                 ;PTR. TO INIT CODE
002104  021326'                        .WORD   L$INIT
002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  022406'                        .WORD   L$CLEAN
002110                          L$ACP::                 ;PTR. TO AUTO CODE
002110  022326'                        .WORD   L$AUTO
002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021316'                        .WORD   L$PROT
002114                          L$TEST::                ;TEST NUMBER
002114  000000                          .WORD   0
002116                          L$DLY::                 ;DELAY COUNT
002116  000000                          .WORD   0
002120                          L$HIME::                ;PTR. TO HIGH MEM
002120  000000                          .WORD   0
34
35
36
37                                      .SBTTL  DISPATCH TABLE
38
```

```
                                        .TITLE  TSV6 - PARAMETER CODING

                                        BGNMOD  TSV6
045442                          TSV6::
045442

                                        .SBTTL  HARDWARE PARAMETER CODING SECTION

                                ;++
                                ; 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.
                                ;--
045442                                  BGNHRD
045442  000010                          .WORD L10107-L$HARD/2
045444                          L$HARD::

045444                                  GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
045444  000031                          .WORD   T$CODE
045446  045464'                         .WORD   HPM1
045450  160010                          .WORD   T$LOLIM
045452  177776                          .WORD   T$HILIM
045454                                  GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
045454  001031                          .WORD   T$CODE
045456  045520'                         .WORD   HPM2
045460  000000                          .WORD   T$LOLIM
045462  000776                          .WORD   T$HILIM
                                        ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
045464                                  ENDHRD
                                        .EVEN
045464                          L10107:
045464  104     105     126     HPM1:   .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
045520  111     116     124     HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
045544  111     116     124     HPM3:   .ASCIZ  'INTERRUPT PRIORITY      '
                                        .EVEN

                                        .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--
045574                                  BGNSFT
045574  000003                          .WORD L10110-L$SOFT/2
045576                          L$SOFT::
                                        GPRMA   SPM1,0,-1,YES                   ; GET TRANSPORT TEST FLAG.
045576                                  GPRMA   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
045576  001130                          .WORD   T$CODE
```

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

AS YOU CAN SEE FROM THE ABOVE EXAMPLE, THE HARDWARE PARAMETERS DO NOT VARY SIGNIFICANTLY FROM UNIT TO UNIT. THE PROCEDURE SHOWN IS NOT VERY EFFICIENT.

THE RUNTIME SERVICES CAN TAKE MULTIPLE UNIT SPECIFICATIONS HOWEVER. LET'S BUILD THE SAME TABLE USING THE MULTIPLE SPECIFICATION FEATURE.

```
# UNITS (D) ?  8<CR>

UNIT 1
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  0,1<CR>
Q-FACTOR (O) O ?  1,0<CR>

UNIT 3
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  2-5<CR>
Q-FACTOR (O) O ?  0<CR>

UNIT 7
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  6,7<CR>
Q-FACTOR (O) O ?  1<CR>
```

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THE FIRST PASS, TWO ENTRIES ARE BUILT SINCE TWO SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE SERVICES ASSUME THAT THE CSR ADDRESS IS 160000 FOR BOTH SINCE IT WAS SPECIFIED ONLY ONCE. IN THE SECOND PASS, FOUR ENTRIES WERE BUILT. THIS IS BECAUSE FOUR SUB-DEVICES WERE SPECIFIED. THE "-" CONSTRUCT TELLS THE RUNTIME SERVICES TO INCREMENT THE DATA FROM THE FIRST NUMBER TO THE SECOND. IN THIS CASE, SUB-DEVICES 2, 3, 4 AND 5 WERE SPECIFIED. (IF THE SUB-DEVICE WERE SPECIFIED BY ADDRESSES, THE INCREMENT WOULD BE BY 2 SINCE ADDRESSES MUST BE ON AN EVEN BOUNDARY.) THE CSR ADDRESSES AND Q-FACTORS FOR THE FOUR ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST TWO UNITS ARE SPECIFIED IN THE THIRD PASS.

THE WHOLE PROCESS COULD HAVE BEEN ACCOMPLISHED IN ONE PASS AS SHOWN BELOW.

```
# UNITS (D) ?  8<CR>

UNIT 1
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  0-7<CR>
Q-FACTOR (O) O ?  0,1,0,,,,1,1<CR>
```

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING

```
                                        .TITLE  TSV3 - GLOBAL AREAS
                                        .SBTTL  GLOBAL EQUATES SECTION


002174                                  BGNMOD  TSV3
002174                          TSV3::


                                        .SBTTL  GLOBAL EQUATES SECTION

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

002174                                  EQUALS          ; GET STANDARD EQUATES.
                                ;
                                ; BIT DIFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
000040                          EF.START==      32.                     ; START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
000036                          EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
000035                          EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
000034                          EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
                                ;
                                ;
```

```
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
000140                  PRI03== 140
000100                  PRI02== 100
000040                  PRI01== 40
000000                  PRI00== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==       4
000010                  LOT==      10
000020                  ADR==      20
000040                  IDU==      40
000100                  ISR==     100
000200                  UAM==     200
000400                  BOE==     400
001000                  PNT==    1000
002000                  PRI==    2000
004000                  IXE==    4000
010000                  IBE==   10000
020000                  IER==   20000
040000                  LOE==   40000
100000                  HOE==  100000
34
35 002174                       KT11    ;;                      ;DEFINE MEMORY MANAGEMENT REGISTERS
                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS
                        ;*KT11 VECTOR ADDRESS
000250                  MMVEC=  250
                        ;*KT11 STATUS REGISTER ADDRESSES
177572                  SR0=    177572
177574                  SR1=    177574
177576                  SR2=    177576
172516                  SR3=    172516
                        .IF NB
                        ;*USER "I" PAGE DESCRIPTOR REGISTERS
                        UIPDR0= 177600
                        UIPDR1= 177602
                        UIPDR2= 177604
                        UIPDR3= 177606
                        UIPDR4= 177610
                        UIPDR5= 177612
                        UIPDR6= 177614
                        UIPDR7= 177616
                        .IF NB
                        ;*USER "D" PAGE DESCRIPTOR REGISTERS
                        UDPDR0= 177620
                        UDPDR1= 177622
                        UDPDR2= 177624
                        UDPDR3= 177626
                        UDPDR4= 177630
                        UDPDR5= 177632
                        UDPDR6= 177634
                        UDPDR7= 177636
```

```
.ENDC
;*USER "I" PAGE ADDRESS REGISTERS
UIPAR0= 177640
UIPAR1= 177642
UIPAR2= 177644
UIPAR3= 177646
UIPAR4= 177650
UIPAR5= 177652
UIPAR6= 177654
UIPAR7= 177656
.IF NB
;*USER "D" PAGE ADDRESS REGISTERS
UDPAR0= 177660
UDPAR1= 177662
UDPAR2= 177664
UDPAR3= 177666
UDPAR4= 177670
UDPAR5= 177672
UDPAR6= 177674
UDPAR7= 177676
.ENDC
.ENDC
.IF NB
;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
SIPDR0= 172200
SIPDR1= 172202
SIPDR2= 172204
SIPDR3= 172206
SIPDR4= 172210
SIPDR5= 172212
SIPDR6= 172214
SIPDR7= 172216
.IF NB
;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
SDPDR0= 172220
SDPDR1= 172222
SDPDR2= 172224
SDPDR3= 172226
SDPDR4= 172230
SDPDR5= 172232
SDPDR6= 172234
SDPDR7= 172236
.ENDC
;*SUPERVISOR "I" PAGE ADDRESS REGISTERS
SIPAR0= 172240
SIPAR1= 172242
SIPAR2= 172244
SIPAR3= 172246
SIPAR4= 172250
SIPAR5= 172252
SIPAR6= 172254
SIPAR7= 172256
.IF NB
;*SUPERVISOR "D" PAGE ADDRESS REGISTERS
SDPAR0= 172260
SDPAR1= 172262
SDPAR2= 172264
```

```
                        SDPAR3= 172266
                        SDPAR4= 172270
                        SDPAR5= 172272
                        SDPAR6= 172274
                        SDPAR7= 172276
                        .ENDC
                        .ENDC
                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
172300                  KIPDR0= 172300
172302                  KIPDR1= 172302
172304                  KIPDR2= 172304
172306                  KIPDR3= 172306
172310                  KIPDR4= 172310
172312                  KIPDR5= 172312
172314                  KIPDR6= 172314
172316                  KIPDR7= 172316
                        .IF NB
                        ;*KERNEL "D" PAGE
                         DESCRIPTOR REGISTERS
                        KDPDR0= 172320
                        KDPDR1= 172322
                        KDPDR2= 172324
                        KDPDR3= 172326
                        KDPDR4= 172330
                        KDPDR5= 172332
                        KDPDR6= 172334
                        KDPDR7= 172336
                        .ENDC
                        ;*KERNEL "I" PAGE ADDRESS REGISTERS
172340                  KIPAR0= 172340
172342                  KIPAR1= 172342
172344                  KIPAR2= 172344
172346                  KIPAR3= 172346
172350                  KIPAR4= 172350
172352                  KIPAR5= 172352
172354                  KIPAR6= 172354
172356                  KIPAR7= 172356
                        .IF NB
                        ;*KERNEL "D" PAGE ADDRESS REGISTERS
                        KDPAR0= 172360
                        KDPAR1= 172362
                        KDPAR2= 172364
                        KDPAR3= 172366
                        KDPAR4= 172370
                        KDPAR5= 172372
                        KDPAR6= 172374
                        KDPAR7= 172376
                        .ENDC
     39
     40
     41                         .SBTTL  TSU05 REGISTER AND PACKET DEFINITIONS
     42
     43                 ;
     44                 ; SOME GENERAL EQUATES.
     45                 ;
     46
     47 000004          ERRVEC==        4               ; POINTER TO ERROR VECTOR FOR BU  . ' OUT.
```

FOLLOWS:

CHANGE SW (L) ?

INHIBIT ITERATIONS (L) N ?

6.0 TEST SUMMARIES

TEST 1: BUS RESET TEST

THIS TEST VERIFIES THAT THE M7455 MODULE'S DEVICE REGISTERS ARE ACCESSIBLE ON THE BUS (SUBTEST 1) AND THEN CHECKS THAT THE BUILT-IN INITIALIZATION SELF-TEST MICRODIAGNOSTIC DID NOT FIND ANY BASIC PROBLEMS IN THE MODULE. AREAS OF LOGIC TESTED BY THE SELF-TEST SEQUENCE ARE AS FOLLOWS: ROM AND PIPELINE REGISTER, SEQUENCER, INTERNAL BUSES, 2901 MICROPROCESSOR, AND, RAM. THIS TEST INITIALIZES THE CONTROLLER BY ISSUING THE BUS INIT SIGNAL VIA A RESET INSTRUCTION, OR BY WRITING INTO THE TSSR REGISTER, WAITS A PERIOD OF TIME (TO ALLOW THE CONTROLLER'S INITIALIZATION MICRODIAGNOSTIC SEQUENCE TO BE COMPLETED), AND THEN CHECKS THE CONTENTS OF THE TSSR REGISTER. SUCCESSFUL INITIALIZATION IS INDICATED BY SUBSYSTEM READY (SSR) AND NEED BUFFER ADDRESS (NBA) BITS BEING SET (1) AND ALL OTHER BITS (EXCEPT A17 AND A16 AND OFL, WHICH ARE IGNORED FOR THIS TEST) BEING CLEAR (0). IF THE CONTENTS OF TSSR ARE NOT AS EXPECTED, AN ERROR REPORT IS ISSUED LISTING THE EXPECTED DATA, ACTUAL DATA, AND THE DISCREPANCIES. THE ERROR REPORT ANALYZES THE TSSR CONTENTS AND DISCERNS AND REPORTS ONE OF THREE POSSIBILITIES:

1. TSSR CONTENTS ARE AMBIGUOUS (ANY OF BITS 11-14 ARE SET, OR STATES OF SSR AND SC BITS DO NOT CORRESPOND TO THE APPARENT ERROR CODE IN BITS 0-5): INDICATES THAT THE TSSR CONTENT CANNOT BE TRUSTED. INDICATES A CATASTROPHIC CONTROLLER MALFUNCTION. THIS IS A FATAL ERROR (EXECUTION IS ABORTED). FIELD ACTION WOULD BE TO REPLACE THE M7455. IF THE M7455 ITSELF IS BEING DEBUGGED, THE PROGRAM SHOULD BE RESTARTED WITH LOOP ON ERROR ENABLED IN ORDER TO PROBE FOR THE PROBLEM.

2. SSR = 0, SC = 0 AND THE ERROR CODE IN BITS 0-5 IS IN THE RANGE 17-13: THIS IS A FATAL ERROR. THE ERROR CODE IS DECODED AND THE APPROPRIATE DESCRIPTION GIVEN. INDICATES THAT A SERIOUS PROBLEM EXISTS.

TEST 2: WRAP DATA - HIGH BYTE

THIS TEST VERIFIES OPERATION OF:

1. PART OF THE PDP-11 BUS INTERFACE SECTION OF THE M7455 MODULE: PART OF THE INPUT FILE (TSDB HIGH BYTE), PART

OF THE OUTPUT FILE (TSSR HIGH BYTE AND TSBA, BOTH BYTES), PART OF THE DC005 TRANSCEIVER CIRCUITS (ADDRESS DECODER, BDAL DRIVERS, HIGH BYTE OF INTERNAL DAL BUS DRIVERS), AND BASIC PROGRAMMED I/O CONTROL SEQUENCES AND LOGIC;

2. PART OF 2901 MICROPROCESSOR ELEMENTS (Q-REGISTER, REGISTER 0, ROTATE AND NEGATE FUNCTIONS

3. Y AND SOURCE BUSES;

4. BASIC MICROPROGRAM SEQUENCES.

THE PROGRAM WRITES A TEST DATA BYTE INTO THE HIGH BYTE OF TSDB, WAITS FOR THE SSR BIT IN TSSR TO SET, THEN CHECKS THE CONTENTS OF BOTH TSBA AND TSSR. THE MODULE IS FUNCTIONING CORRECTLY IF DATA WRITTEN APPEARS IN BOTH BYTES OF TSBA AND THE FINAL CONTENT OF TSSR IS CORRECT (SAME AS AFTER INITIALIZATION EXCEPT FOR BITS 8 AND 9, WHICH SHOULD CONTAIN BITS 8 AND 9 OF THE DATA PATTERN WRITTEN. AN ERROR IS REPORTED AND A DESCRIPTIVE ANALYSIS GIVEN IF A DISCREPANCY IN TSBA OR TSSR IS DETECTED. THE ANALYSIS LISTS LIKELY FAULTY CANDIDATES FROM THE LOGIC ELEMENTS LISTED ABOVE. THE TEST IS REPEATED FOR ALL COMBINATIONS OF TEST DATA BYTES (0-377 OCTAL).

TEST 3: WRAP DATA - LOW BYTE

THIS TEST FURTHER VERIFIES OPERATION OF MANY OF THE SAME ELEMENTS TESTED IN TEST 2, AND ADDITIONALLY VERIFIES:

1. LOW BYTE OF THE TSDB INPUT FILE REGISTER,

2. LOW BYTE OF INTERNAL DAL BUS DRIVERS ON THE DC005 TRANSCEIVER CIRCUITS,

3. BASIC FUNCTIONING OF PARTS OF THE RAM.

THE PROGRAM WRITES A TEST DATA BYTE INTO THE LOW BYTE OF TSDB, WAITS FOR THE SSR BIT IN TSSR TO SET, THEN CHECKS THE CONTENTS OF BOTH TSBA AND TSSR. THE MODULE IS FUNCTIONING CORRECTLY IF DATA WRITTEN APPEARS IN BOTH BYTES OF TSBA AND THE FINAL CONTENT OF TSSR IS CORRECT (SAME AS AFTER INITIALIZATION EXCEPT FOR BITS 8 AND 9, WHICH SHOULD CONTAIN BITS 8 AND 9 OF THE DATA PATTERN WRITTEN. AN ERROR IS REPORTED AND A DESCRIPTIVE ANALYSIS GIVEN IF A DISCREPANCY IN TSBA OR TSSR IS DETECTED. THE ANALYSIS LISTS LIKELY FAULTY CANDIDATES FROM THE LOGIC ELEMENTS LISTED ABOVE. THE TEST IS REPEATED FOR ALL COMBINATIONS OF TEST DATA BYTES (0-377 OCTAL).

TEST 4: RAM TEST

THIS TEST VERIFIES THAT ALL LOCATIONS OF THE RAM ON THE M7455 CAN PROPERLY STORE AND READ BACK ALL DATA PATTERNS, AND THAT EACH RAM LOCATION IS UNIQUELY ADDRESSED (I.E., THAT ONE AND ONLY ONE LOCATION IS ACCESSED BY ANY PARTICULAR ADDRESS). THE BYPRODUCT OF THESE TESTS IS A VERIFICATION OF TWO REGISTERS IN THE 2901 AND THE CAPABLITY OF THE 2901 TO CORRECTLY PERFORM AN ADD.

TEST 5: SECOND INITIALIZATION TEST

THIS TEST VERIFIES THE SAME ELEMENTS AS DID INITIALIZATION TEST #1 AND ALSO CHECKS THAT CERTAIN PARTS OF RAM IS CLEARED TO ZERO AND THAT 2901 REGISTERS 10 AND 11 ARE ALSO CLEARED TO ZERO. THIS IS A CONFIDENCE CHECK OF A PART OF THE SELF-TEST SEQUENCE (I.E., THAT IT IS REALLY BEING EXECUTED). FOR EACH OF TWO SUBTESTS (ONE FOR INITIALIZING VIA A BUS INIT, THE OTHER FOR INITIALIZING BY WRITING INTO THE TSSR), THE FOLLOWING SEQUENCE IS PERFORMED:

1. EACH RAM LOCATION AND 2901 REGISTERS 10 AND 11 ARE SET TO ALL 1'S BY USING WRITES INTO THE TSDB REGISTER (LOW BYTE AND MAINTENANCE MODE WORD WRITES).

2. THE CONTROLLER IS INITIALIZED AND THE VARIOUS CHECKS ON THE TSSR DESCRIBED IN INITIALIZATION TEST #1 ARE PERFORMED.

3. #1'S (377 OCTAL) ARE WRITTEN INTO THE LOW BYTE OF TSDB, WHICH SHOULD CAUSE RAM LOCATION 0 TO BE WRITTEN TO ALL 1'S SINCE 2901 REGISTERS 10 AND 11, SPECIFYING THE RAM ADDRESS, SHOULD BE 0. RAM LOCATION 0 IS VERIFIED BY WRITING A WORD OF ZEROS INTO THE TSDB. THE RESULTING LOW BYTE OF TSBA SHOULD CONTAIN ALL 1'S.

4. THE ENTIRE RAM IS SCANNED. LOCATION 0 SHOULD CONTAIN ALL 1'S AND THE REMAINING LOCATIONS, EXCEPT FOR THE MESSAGE BUFFER IMAGE AREA, SHOULD CONTAIN 0. DISCREPANCIES ARE REPORTED. AN ERROR AT THIS POINT IS MOST LIKELY DUE TO A ROM, PIPELINE OR SEQUENCER PROBLEM OR A TIMING PROBLEM.

TEST 6: COMMAND REJECT

THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA AND TSSR REGISTERS ARE LEFT IN THE PROPER STATE AFTER EACH COMMAND IS REJECTED. THIS TEST CHECKS MICROPROCESSOR SEQUENCING, BASIC COMMAND DECODING AND DATI DMA HANDLING. THIS TEST CONTAINS TWO SUBTESTS: SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER THAN WRITE CHARACTERISTICS) WITH THE INTERRUPT ENABLE (IE) BIT CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE

AND THE REGISTER MODIFICATION REFUSED (RMR) LOGIC.

TEST 11: NON-TAPE MOTION COMMANDS

THIS TEST VERIFIES PROPER OPERATION OF THE INITIALIZE COMMAND. TWO SUBTESTS ARE USED. THE FIRST VERIFIES THAT THE COMMAND RUNS TO COMPLETION AND STORES A VALID MESSAGE PACKET. THE SECOND VERIFIES THAT NON-ZERO VALUES IN THE COMMAND MODE FIELD CAUSES COMMAND REJECT.

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

```
                                        .TITLE  TSV2 - PROGRAM HEADER
                                        .SBTTL  PROGRAM HEADER
000000                                  .PSECT  ABS

                                        .MCALL  SVC
000000                                  SVC                     ; INITIALIZE SUPERVISOR MACROS
                                        .ENABLE LC
                                        .NLIST  BEX,CND
                                        .ENABL  AMA
        002000'                         .=.+2000
                                ;       .=2000
002000                                  BGNMOD  TSV2
002000                          TSV2::

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


002000                                  POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
002000                                  HEADER  CZTSA,A,0,655.,0
002000                          L$NAME::                ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  132                             .ASCII /Z/
002002  124                             .ASCII /T/
002003  123                             .ASCII /S/
002004  101                             .ASCII /A/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                          L$REV::                 ;REVISION LEVEL
002010  101                             .ASCII  /A/
002011                          L$DEPO::                ;0
002011  060                             .ASCII  /0/
002012                          L$UNIT::                ;NUMBER OF UNITS
002012  000000                          .WORD   0
002014                          L$TIML::                ;LONGEST TEST TIME
002014  001217                          .WORD   655.
002016                          L$HPCP::                ;POINTER TO H.W. QUES.
002016  045444'                         .WORD   L$HARD
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  045576'                         .WORD   L$SOFT
002022                          L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002154'                         .WORD   L$HW
002024                          L$SPTP::                ;PTR. TO S.W. PTABLE
002024  002164'                         .WORD   L$SW
002026                          L$LADP::                ;DIAG. END ADDRESS
002026  045672'                         .WORD   L$LAST
002030                          L$STA::                 ;RESERVED FOR APT STATS
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                          .WORD   0
```

PROGRAM HEADER

```
002036                          L$APT::                 ;APT EXPANSION
002036  000000                          .WORD   0
002040                          L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002124'                        .WORD   L$DISPATCH
002042                          L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000000                          .WORD   0
002044                          L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
002044  000000                          .WORD   0
002046                          L$EXP1::                ;EXPANSION WORD
002046  000000                          .WORD   0
002050                          L$MREV::                ;SVC REV AND EDIT #
002050     003                          .BYTE   C$REVISION
002051     003                          .BYTE   C$EDIT
002052                          L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                          .WORD   0
002054  000000                          .WORD   0
002056                          L$SPC::
002056  000000                          .WORD   0
002060                          L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  003400'                        .WORD   L$DVTYP
002062                          L$REPP::                ;PTR. TO REPORT CODE
002062  022434'                        .WORD   L$RPT
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  022122'                        .WORD   L$AU
002072                          L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022220'                        .WORD   L$DU
002074                          L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                          .WORD   0
002076                          L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  003406'                        .WORD   L$DESC
002100                          L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                        EMT     E$LOAD
002102                          L$ETP::                 ;POINTER TO ERRTBL
002102  000000                          .WORD   0
002104                          L$ICP::                 ;PTR. TO INIT CODE
002104  021326'                        .WORD   L$INIT
002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  022406'                        .WORD   L$CLEAN
002110                          L$ACP::                 ;PTR. TO AUTO CODE
002110  022326'                        .WORD   L$AUTO
002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021316'                        .WORD   L$PROT
002114                          L$TEST::                ;TEST NUMBER
002114  000000                          .WORD   0
002116                          L$DLY::                 ;DELAY COUNT
002116  000000                          .WORD   0
002120                          L$HIME::                ;PTR. TO HIGH MEM
002120  000000                          .WORD   0
34
35
36
37                                      .SBTTL  DISPATCH TABLE
38
```

J2


```
     76 002172  000310                GERRMAX::       .WORD   200.    ; GLOBAL (PER UNIT) ERROR LIMIT
     77 002174                                ENDSW
        002174                          L10001:
     78
     79 002174                                ENDMOD
     80
     81
     84
     85
```

```
                                        .TITLE  TSV3 - GLOBAL AREAS
                                        .SBTTL  GLOBAL EQUATES SECTION


002174                                  BGNMOD  TSV3
002174                          TSV3::


                                        .SBTTL  GLOBAL EQUATES SECTION

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

002174                                  EQUALS          ; GET STANDARD EQUATES.
                                ;
                                ; BIT DIFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
000040                          EF.START==      32.             ; START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.             ; RESTART COMMAND WAS ISSUED
000036                          EF.CONTINUE==   30.             ; CONTINUE COMMAND WAS ISSUED
000035                          EF.NEW==        29.             ; A NEW PASS HAS BEEN STARTED
000034                          EF.PWR==        28.             ; A POWER-FAIL/POWER-UP OCCURRED
                                ;
                                ;
```

```
                                ; PRIORITY LEVEL DEFINITIONS
                                ;
000340                          PRI7== 340
000300                          PRI6== 300
000240                          PRI5== 240
000200                          PRI4== 200
000140                          PRI3== 140
000100                          PRI2== 100
000040                          PRI1== 40
000000                          PRI0== 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
34
35 002174                               KT11    ,,                      ;DEFINE MEMORY MANAGEMENT REGISTERS
                                .SBTTL  MEMORY MANAGEMENT DEFINITIONS
                                ;*KT11 VECTOR ADDRESS
000250                          MMVEC=  250
                                ;*KT11 STATUS REGISTER ADDRESSES
177572                          SR0=    177572
177574                          SR1=    177574
177576                          SR2=    177576
172516                          SR3=    172516
                                .IF NB
                                ;*USER "I" PAGE DESCRIPTOR REGISTERS
                                UIPDR0= 177600
                                UIPDR1= 177602
                                UIPDR2= 177604
                                UIPDR3= 177606
                                UIPDR4= 177610
                                UIPDR5= 177612
                                UIPDR6= 177614
                                UIPDR7= 177616
                                .IF NB
                                ;*USER "D" PAGE DESCRIPTOR REGISTERS
                                UDPDR0= 177620
                                UDPDR1= 177622
                                UDPDR2= 177624
                                UDPDR3= 177626
                                UDPDR4= 177630
                                UDPDR5= 177632
                                UDPDR6= 177634
                                UDPDR7= 177636
```

```
.ENDC
;*USER "I" PAGE ADDRESS REGISTERS
UIPAR0= 177640
UIPAR1= 177642
UIPAR2= 177644
UIPAR3= 177646
UIPAR4= 177650
UIPAR5= 177652
UIPAR6= 177654
UIPAR7= 177656
.IF NB
;*USER "D" PAGE ADDRESS REGISTERS
UDPAR0= 177660
UDPAR1= 177662
UDPAR2= 177664
UDPAR3= 177666
UDPAR4= 177670
UDPAR5= 177672
UDPAR6= 177674
UDPAR7= 177676
.ENDC
.ENDC
.IF NB
;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
SIPDR0= 172200
SIPDR1= 172202
SIPDR2= 172204
SIPDR3= 172206
SIPDR4= 172210
SIPDR5= 172212
SIPDR6= 172214
SIPDR7= 172216
.IF NB
;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
SDPDR0= 172220
SDPDR1= 172222
SDPDR2= 172224
SDPDR3= 172226
SDPDR4= 172230
SDPDR5= 172232
SDPDR6= 172234
SDPDR7= 172236
.ENDC
;*SUPERVISOR "I" PAGE ADDRESS REGISTERS
SIPAR0= 172240
SIPAR1= 172242
SIPAR2= 172244
SIPAR3= 172246
SIPAR4= 172250
SIPAR5= 172252
SIPAR6= 172254
SIPAR7= 172256
.IF NB
;*SUPERVISOR "D" PAGE ADDRESS REGISTERS
SDPAR0= 172260
SDPAR1= 172262
SDPAR2= 172264
```

```
                        SDPAR3= 172266
                        SDPAR4= 172270
                        SDPAR5= 172272
                        SDPAR6= 172274
                        SDPAR7= 172276
                        .ENDC
                        .ENDC
                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
172300                  KIPDR0= 172300
172302                  KIPDR1= 172302
172304                  KIPDR2= 172304
172306                  KIPDR3= 172306
172310                  KIPDR4= 172310
172312                  KIPDR5= 172312
172314                  KIPDR6= 172314
172316                  KIPDR7= 172316
                        .IF NB
                        ;*KERNEL "D" PAGE
                         DESCRIPTOR REGISTERS
                        KDPDR0= 172320
                        KDPDR1= 172322
                        KDPDR2= 172324
                        KDPDR3= 172326
                        KDPDR4= 172330
                        KDPDR5= 172332
                        KDPDR6= 172334
                        KDPDR7= 172336
                        .ENDC
                        ;*KERNEL "I" PAGE ADDRESS REGISTERS
172340                  KIPAR0= 172340
172342                  KIPAR1= 172342
172344                  KIPAR2= 172344
172346                  KIPAR3= 172346
172350                  KIPAR4= 172350
172352                  KIPAR5= 172352
172354                  KIPAR6= 172354
172356                  KIPAR7= 172356
                        .IF NB
                        ;*KERNEL "D" PAGE ADDRESS REGISTERS
                        KDPAR0= 172360
                        KDPAR1= 172362
                        KDPAR2= 172364
                        KDPAR3= 172366
                        KDPAR4= 172370
                        KDPAR5= 172372
                        KDPAR6= 172374
                        KDPAR7= 172376
                        .ENDC
39
40
41                              .SBTTL  TSU05 REGISTER AND PACKET DEFINITIONS
42
43                      ;
44                      ; SOME GENERAL EQUATES.
45                      ;
46
47      000004          ERRVEC==        4               ; POINTER TO ERROR VECTOR FOR BU.  . ' UUI.
```

TSU05 REGISTER AND PACKET DEFINITIONS

```
000060                  TTIVEC==        60              ; INTERRUPT VECTOR FOR CONSOLE INPUT
177560                  TTICSR==        177560          ; BUS ADDRESS OF CONSOLE INPUT
177562                  TTIBFR==        177562          ; CONSOLE INPUT DATA BUFFER
177520                  BDVPCR==        177520          ; BDV11 PAGE CONTROL REGISTER

                        ;*
                        ;BIT DEFINITIONS FOR TSSR REGISTER
                        ;-

100000                  SC=     BIT15           ;SPECIAL CONDITION
040000                  BIE=    BIT14           ;BUS INTERFACE ERROR
020000                  SCE=    BIT13           ;SANITY CHECK ERROR
010000                  RMR=    BIT12           ;MODIFICATION REFUSED
004000                  NXM=    BIT11           ;NONEXISTANT MEMORY ERROR
002000                  NBA=    BIT10           ;NEED BUFFER ADDRESS
001400                  HIADDR= BIT9!BIT8       ;EXTENDED ADDRESS BITS
000200                  SSR=    BIT7            ;SUB SYSTEM READY
000100                  OFL=    BIT6            ;OFF LINE BIT
000060                  FATERR= BIT4!BIT5       ;FATAL TERMINATION ERROR CODES
000016                  TERCLS= BIT3!BIT2!BIT1  ;TERMINATION CODES


                        ;*
                        ;
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
                        ;(XST0)
                        ;
                        ;-

100000                  XSOTMK= BIT15           ;TAPE MARK DETECTED
040000                  XSORLS= BIT14           ;RECORD LENGTH SHORT
020000                  XSOLET= BIT13           ;LOGICAL END OF TAPE
010000                  XSORLL= BIT12           ;RECORD LENGTH LONG
004000                  XSOWLE= BIT11           ;WRITE LOCK ERROR
002000                  XSONEF= BIT10           ;NON EXECUTABLE FUNCTION
001000                  XSOILC= BIT9            ;ILLEGAL COMMAND
000400                  XSOILA= BIT8            ;ILLEGAL ADDRESS
000200                  XSOMOT= BIT7            ;TAPE IN MOTION
000100                  XSOONL= BIT6            ;TRANSPORT ON LINE
000040                  XSOIE=  BIT5            ;INTERRUPT ENABLE
000020                  XSOVCK= BIT4            ;VOLUME CHECK BIT
000010                  XSOPED= BIT3            ;PHASE ENCODED DRIVE
000004                  XSOWLK= BIT2            ;WRITE LOCKED
000002                  XSOBOT= BIT1            ;BEGINNING OF TAPE
000001                  XSOEOT= BIT0            ;END OF TAPE


                        ;*
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
                        ;(XST1)
                        ;-

100000                  X1.DLT  = BIT15         ;DATA LATE
040000                  X1.SPARE= BIT14         ;NOT USED
020000                  X1.COR  = BIT13         ;CORRECTABLE DATA ERROR
017375                  X1.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
000400                  X1.RBP  = BIT8          ;READ BUS PARITY ERROR
000002                  X1.UNC  = BIT1          ;UNCORRECTABLE DATA OR HARD ERROR
```

```
                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
                        ;(XST2)
                        ;-
100000                  X2.OPM  = BIT15         ;OPERATION IN PROGRESS (TAPE MOVING)
040000                  X2.RCE  = BIT14         ;RAM CHECKSUM ERROR
035400                  X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8   ;NOT USED BY TSU'S (ALWAYS=0)
002000                  X2.WCF  = BIT10         ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
000200                  X2.EXTF = BIT7          ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
000100                  X2.BUFE = BIT6          ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
000077                  X2.REV  = 000077        ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
000007                  X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
                        ;(XST3)
                        ;-
177400                  X3.MDE  = 177400        ;MICRO-DIAGNOSTIC ERROR CODE
000200                  X3.SPARE= BIT7          ;NOT USED BY TSU05
000100                  X3.OPI  = BIT6          ;OPERATION INCOMPLETE
000040                  X3.REV  = BIT5          ;REVERSE
000020                  X3.TRF  = BIT4          ;TRANSPORT RESPONSE FAILURE
000010                  X3.DCK  = BIT3          ;DENSITY CHECK
000006                  X3.MBZ  =BIT2+BIT1      ;NOT USED ALWAYS 0
000001                  X3.RIB  = BIT0          ;REVERSE INTO BOT

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
                        ;(XST4)
                        ;-
100000                  X4.HSP  = BIT15         ;HIGH SPEED
040000                  X4.RCE  = BIT14         ;RETRY COUNT EXCEEDED
020000                  X4.TSM  = BIT13         ;TRANSPORT SPECIAL MODE
017400                  X4.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT8   ;NOT USED ALWAYS 0
000377                  X4.WRC  = 000377        ;WRITE RETRY COUNT FIELD


                        ;+
                        ;
                        ;TSSR TERMINATION CODES (BIT 0-2)
                        ;
                        ;-

000006                          TSREJ=  3+2             ;COMMAND REJECTED
000006                          UNREC=  6               ;UNRECOVERABLE ERROR

                        ;+
                        ;
                        ;DEVICE REGISTER OFFSETS
                        ;
                        ;-

000000                  TSBA==  0
000000                  TSDB==  0               ;TSDB/TSBA REGISTER
000001                  TSBAH== 1
000001                  TSDBH== 1               ;TSDB/TSBA REGISTER HIGH BYTE
```

```
                        ;+
                        ; TSDB ADDRESS BIT DEFINITIONS
                        ;-
000003                  A1716   = BIT1+BIT0     ;ADDRESS BITS 17:16 ARE IN 1:0

                        ;+
                        ; COMMAND DEFINITIONS
                        ;-
000017                  P.GETSTAT       = 17    ;GET STATUS
000013                  P.INIT          = 13    ;INITIALIZE
000012                  P.CONTROL       = 12    ;CONTROL COMMANDS
000011                  P.FORMAT        = 11    ;FORMAT
000010                  P.POSITION      = 10    ;POSITION
000006                  P.WRTSUB        = 6     ;SUBSYSTEM WRITE
000005                  P.WRITE         = 5     ;WRITE
000004                  P.WRTCHAR       = 4     ;WRITE CHARACTERISTICS
000001                  P.READ          = 1     ;READ

                        ;+
                        ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
                        ;-
100000                          P.ACK   = BIT15         ;BUFFER AVAIL FOR CONTROLLER
040000                          P.CVC   = BIT14         ;CLEAR VOLUME CHECK
020000                          P.OPP   = BIT13         ;REVERSE SEQUENCE OF DATA BITS
010000                          P.SWB   = BIT12         ;SWAP BYTES IN MEMORY
007400                          P.MODE  = BIT11!BIT10!BIT9!BIT8 ;EXTENDED COMMAND MODE FIELD
000200                          P.IE    = BIT7          ;INTERRUPT ENABLE
000140                          P.FMT= BIT6!BIT5        ;PACKET HEADER TYPE (ALWAYS=0)
000037                          P.CMD   = 37            ;MAJOR COMMAND FIELD

                        ;+
                        ; CONTROL COMMAND MODE CODES
                        ;-
000000                          PC.RELEASE      = 0*256.        ;RELEASE BUFFER
000400                          PC.REWIND       = 1*256.        ;REWIND
001000                          PC.NOOP         = 2*256.        ;NO-OP
002000                          PC.IEREW        = 4*256.        ;REWIND IMMEDIATE INTERRUPT
002400                          PC.ERASE        = 5*256.        ;SECURITY ERASE

                        ;+
                        ; CONTROLLER RAM DEFINITIONS
                        ;-
000167                  RMCHBEG = 167           ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
000200                  RMCHEND = 200           ;CHARACTERISTICS IO DATA END RAM ADDRESS
000201                  RMPKTBEG= 201           ;COMMAND PACKET BEGIN RAM ADDRESS
000210                  RMPKTEND= 210           ;COMMAND PACKET END RAM ADDRESS
000215                  RMMSGBEG= 215           ;MESSAGE BUFFER BEGIN RAM ADDRESS
000234                  RMMSGEND= 234           ;MESSAGE BUFFER END RAM ADDRESS
                        ;+
                        ;
                        ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
                        ;
                        ;-

000006                  XST0==  6               ;EXTENDED STATUS REGISTER 0 (WORD 4)
```

```
                        ;+
                        ;BSEL1 CODES FOR WRITE FORMAT
                        ;-
000200                  WF.IHISP        = BIT7          ;IHISP  - HIGH SPEED
000100                  WF.IWRT         = BIT6          ;IWRT   - WRITE
000040                  WF.IREV         = BIT5          ;IREV   - REVERSE
000020                  WF.IWFM         = BIT4          ;IWFM   - WRITE FILE MARK
000010                  WF.IEDIT        = BIT3          ;IEDIT  - EDIT
000004                  WF.IERASE       = BIT2          ;IERASE - ERASE
000002                  WF.I3RESV       = BIT1          ;IRESV3 - RESERVED #3
000001                  WF.I4RESV       = BIT0          ;IRESV4 - RESERVED #4


                        ;+
                        ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
                        ;-
000200                  MS.EXT          = BIT7          ;INVERT SENSE OF EXTENDED FEATURES SWITCH
000020                  MS.RSFIFO       = BIT4          ;RESET FIFO AND INPUT PARITY ERRORR
000010                  MS.RSTAPE       = BIT3          ;RESET TAPE STATUS IN 2 FLIP-FLOPS
000006                  MS.ATTN         = BIT2!BIT1     ;ATTENTION TRIGGER FIELD
000001                  MS.RSD          = BIT0          ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
                        ;+
                        ; MS.ATTN SUBCODES
                        ;-
000000                          MSA.NOP = 0^2           ;NO-OP (NOTHING TRIGGERED)
000002                          MSA.VOL = 1^2           ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
000004                          MSA.NRAM= 2^2           ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
000006                          MSA.FRAM= 3^2           ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
                        ;+
                        ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
                        ;-
000200                  NP.IR           = BIT7          ;INTERRUPT REQUEST (0-1 TRANSITION)
000100                  NP.OUT          = BIT6          ;TAPE DATA DIRECTION OUT (0= IN)
000040                  NP.LOOP         = BIT5          ;ENABLE TRANSPORT LOOPBACK
000020                  NP.WRP          = BIT4          ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
                        ;+
                        ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
                        ;-

000200                  S2.DIM          = BIT7          ;WORD #9 BYTE 2 DATA IN MISS
000100                  S2.ILW          = BIT6          ;               ILLW H
000040                  S2.OUTRDY       = BIT5          ;               OUT RDY H
000020                  S2.INRDY        = BIT4          ;               IN RDY H
000010                  S2.ATIMR        = BIT3          ;               TIMER A FLAG H
000004                  S2.BTIMR        = BIT2          ;               TIMER B FLAG H
000003                  S2.UNDEF        = BIT1+BIT0     ;(UNDEFINED)
100000                  S1.PARIN        = BIT15         ;WORD #8 BYTE 1 PARIN H
040000                  S1.I2RESV       = BIT14         ;               IRESV2
020000                  S1.I1RESV       = BIT13         ;               IRESV1
010000                  S1.IEOT         = BIT12         ;               IEOT L
004000                  S1.IIDENT       = BIT11         ;               IIDENT H
002000                  S1.ICER         = BIT10         ;               ICER H
001000                  S1.IFMK         = BIT9          ;               IFMK H
000400                  S1.IHER         = BIT8          ;               IHER H
000200                  S0.ISPEED       = BIT7          ;WORD #8 BYTE 0 ISPEED H
000100                  S0.IRDY         = BIT6          ;               IRDY L
000040                  S0.IONL         = BIT5          ;               IONL L
```

```
000020          SO.ILDP     = BIT4      ;       ILDP L
000010          SO.IDBY     = BIT3      ;       IDBY L
000004          SO.IRWD     = BIT2      ;       IRWD L
000002          SO.IFBY     = BIT1      ;       IFBY L
000001          SO.IFPT     = BIT0      ;       IFPT L
                ;+
                ;UNIBUS MAP DEFINATIONS
                ;-
170200          MMR0=   170200


                        .SBTTL  SPECIAL MACROS AND OPDEFS.


                ;+
                ;SAVE GENERAL REGS 1 TO 5
                ;-

                        .MACRO  SAVREG
                        JSR     R5,REGSAV
                        .ENDM

                ;+
                ; MACRO TO FORCE AN ERROR
                ;-
                .MACRO  FORCERROR       TAG,NOTSSR
                .NLIST
                .IIF NDF LISTALL, .NLIST
                .LIST
                .IF B NOTSSR
                        MOV     TSSR(R5),R1     ;READ TSSR
                .ENDC
                        MOV     FORCER,FORCER   ;IS FORCER SET? (LEAVE C BIT ALONE)
                        BNE     TAG             ;BR IF YES
                .NLIST
                .IIF NDF LISTALL, .LIST
                .LIST
                .ENDM

                ;+
                ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
                ;       WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
                ;       SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
                ;       FORCER TO 177777
                ;       TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
                ;-
                .MACRO  FORCEXIT        TAG
                .NLIST
                .IIF NDF LISTALL, .NLIST
                .LIST
                        MOV     FORCER,FORCER   ;IS FORCER NEGATIVE?
                        BMI     TAG             ;BR IF YES
                .NLIST
                .IIF NDF LISTALL, .LIST
                .LIST
                .ENDM
                ;+
```

```
                                ; MACRO TO INCREMENT ERROR COUNTS
                                ;-
                                .MACRO  NEXT.ERRNO
                                .NLIST
                                ;;;.IIF NDF LISTALL, .NLIST
                                        ERRNO=ERRNO+1
                                ;;;.IIF NDF LISTALL, .LIST
                                .LIST
                                .ENDM

                                ;+
                                ;MACRO TO PERFORM XOR
                                ;-

                                        .MACRO  XOR     A,B
                                        MOV     A,-(SP)
                                        BIC     B,(SP)
                                        BIC     A,3
                                        BIS     (SP)+,B
                                        .ENDM

000000                                  EN=0                    ; INITIALIZE ERROR NUMBER
                                        .SBTTL  FORCER  - FORCE ERROR FLAG


                                ;
                                ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
                                ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
                                ;

002174  000000                  FORCER::        0       ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
                                                ; - BY THE MACRO "IFERROR").  AN ERROR NEED NOT -
                                                ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.


                                        .SBTTL  GLOBAL DATA SECTION

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

                                ;
                                ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
                                ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
                                ;
002176  000000                  EPRTSW::        .WORD   0       ;PRINT SWITCH
002200  000000                  UNITN::         .WORD   0       ;UNIT # UNDER TEST.
002202  000000                  QVP::           .WORD   0       ;QUICK VERIFY FLAG.
002204  000000                  CSRADDR::       .WORD   0       ;ADDRESS OF CSR FOR CURRENT DEVICE
002206  000224                  IVEC::          .WORD   224     ;INTERRUPT VECTOR
002210  000200                  IPRI::          .WORD   PRI04   ;INTERRUPT PRIORITY.
002212  000000                  TSTCNT::        .WORD   0       ;NUMBER OF TESTS RUN IN THIS PASS
002214  000000                  LOOPCNT::       .WORD   0       ;REMAINING ITERATION COUNT FOR TEST
002216  000000                  DEVCNT::        .WORD   0       ;NUMBER OF DEVICE UNDER TEST
002220  000000                  FATFLG::        .WORD   0       ;SET IF FATAL ERROR IS DETECTED IN TEST
002222  000000                  INTRECV::       .WORD   0       ;SET IF TAPE INTERRUPT WAS RECEIVED
```

```
504 003020  177773                          .WORD   †CBIT2
505 003022  177767                          .WORD   †CBIT3
506 003024  177757                          .WORD   †CBIT4
507 003026  177737                          .WORD   †CBIT5
508 003030  177677                          .WORD   †CBIT6
509 003032  177577                          .WORD   †CBIT7
510 003034  177377                          .WORD   †CBIT8
511 003036  176777                          .WORD   †CBIT9
512 003040  175777                          .WORD   †CBIT10
513 003042  173777                          .WORD   †CBIT11
514 003044  167777                          .WORD   †CBIT12
515 003046  157777                          .WORD   †CBIT13
516 003050  137777                          .WORD   †CBIT14
517 003052  077777                          .WORD   †CBIT15
518 003054  125252                          .WORD   125252          ;ALTERNATING ONES, ZEROS
519 003056  052525                          .WORD   052525          ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
520         003060'                 TBLEND=.
521
522
523                                         .SBTTL  GLOBAL ENVIRONMENT STORAGE
524                                 ;
525                                 ;STORAGE FOR DEVICE REGISTERS
526                                 ;
527 003060  000000  100000  000000  DUMMY:  0,100000,0,0     ;DUMMY DEVICE REGISTERS...
528 003070  000000  000000  000000          0,0,0,0,0,0,0,0
529                                                         ;...FOR MULTI-UNIT CHECKOUT.
530
531
532 003110  000000                  DUFLG::         .WORD   0               ;"DROPPED UNIT" FLAG.
533                                                                         ;INH1BITS CODE IN "CLEAN-UP".
534 003112  000000                  NODEV::         .WORD   0               ;FLAG TO SAY NO DEVICE.
535
536 003114  000000                  TEMP1::         .WORD   0               ;SOME TEMP LOCATIONS.
537 003116  000000                  TEMP2::         .WORD   0
538 003120  000000                  XXCOMM::        .WORD   0               ;XXDP+ COMM BLOCK POINTER.
539 003122  000000                  FREE::          .WORD   0               ;1ST FREE MEMORY ADDRESS...
540 003124  000000                  FRESIZ::        .WORD   0               ;...AND SIZE (IN WORDS).
541 003126  000000                  FREEHI: .WORD   0               ;LAST WORD IN FREE SPACE
542 003130  000000                  KTFLG::         .WORD   0               ;KT11, MEM AVAIL FLAG -
543                                                                         ;- .WORD        0 = <24K OR NO KT -
544                                                                         ;- NZ = >24K AND KT.
545 003132  000000                  KTENABLE::      .WORD   0               ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
546 003134  000000                  NXMFLG::        .WORD   0               ;SET IF WE CAN TEST CLEARED OTHERWISE
547 003136  000000                  NXMLO::         .WORD   0               ;NXM LO ADDRESS BITS
548 003140  000000                  NXMHI::         .WORD   0               ;NXM HI ADDRESS BITS FOR DAL'S 16-21
549 003142  000000                  T23A::          .WORD   0               ;PROCESSOR TYPE FLAG
550 003144  000000                  T23B::          .WORD   0               ;PROCESSOR TYPE FLAG B
551 003146  000000                  T3BFLG::        .WORD   0               ;TEST 3B FLAG †0
552 003150  002000                  PST32W::        .WORD   2000            ;32W BLOCK ADDRESS FOR 32K START
553 003152  000000                  SIFLAG::        .WORD   0               ;
554 003154  000000                  BADDAT::        .WORD   0               ;ACTUAL DATA
555 003156  000000                  GDDAT::         .WORD   0               ;EXPECTED DATA
556 003160  000000                  LOOPFL::        .WORD   0
557 003162                          CTAB::                                  ;CONFIGURATION TABLES.
558 003162  000000                  CTABM::         .WORD   0               ;CONFIG WORK.
559 003164  000000                                  .WORD   0
560 003166  000000                                  .WORD   0
```

```
003170  000000                          .WORD   0
003172  177777                          .WORD   -1              ;END OF MEM TABLE.
003174                          CTABE::
                                ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
                                ;
                                ;       0       =       UNIT NOT TESTED
                                ;       100000  =       UNIT ONLINE, NO ERRORS
                                ;       10XXXX  =       UNIT ONLINE, ENCOUNTERED XXXX ERRORS
                                ;       160000  =       UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
                                ;       160001  =       UNIT DROPPED, NOT IDLE AT START
                                ;       14XXXX  =       UNIT DROPPED, ENCOUNTERED XXXX ERRORS
                                ;
003174                          ERTABL:         .BLKW   64.
003374  000000                  ERTABE:         .WORD   0

003376  000000                  SKIPT:  .WORD   0               ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

                                        .SBTTL  GLOBAL TEXT MESSAGES
                                ;++
                                ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                ; MORE THAN ONE TEST.
                                ;--


                                ;+
                                ;NAMES OF DEVICES SUPPORTED
                                ;-

                                        DEVTYP  <TSU05>
003400                          L$DVTYP::
003400      124     123     125         .ASCIZ  /TSU05/
                                        .EVEN


                                ;+
                                ;TEST DESCRIPTION
                                ;-
                                        DESCRIPT <**** TSU05 DIAG PART 1 - REPLACE M7455 IF ERROR ****>
003406                          L$DESC::
003406      052     052     052         .ASCIZ  /**** TSU05 DIAG PART 1 - REPLACE M7455 IF ERROR ****/
                                        .EVEN


                                ;+
                                ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
                                ;-

003474  003534' 003537' 003543' TSSRBIT::       .WORD   1$,2$,3$,4$,5$,6$,7$,8$
003514  003575' 003601' 003605'                 .WORD   9$,10$,11$,12$,13$,14$,15$,16$
003534     123     103     000  1$:     .ASCIZ  'SC'
003537     102     111     105  2$:     .ASCIZ  'BIE'
003543     123     103     105  3$:     .ASCIZ  'SCE'
003547     122     115     122  4$:     .ASCIZ  'RMR'
003553     116     130     115  5$:     .ASCIZ  'NXM'
003557     116     102     101  6$:     .ASCIZ  'NBA'
```

```
                633 003563      102     111     124  7$:     .ASCIZ  'BIT9'
                634 003570      102     111     124  8$:     .ASCIZ  'BIT8'
                635 003575      123     123     122  9$:     .ASCIZ  'SSR'
                636 003601      117     106     114  10$:    .ASCIZ  'OFL'
                637 003605      102     111     124  11$:    .ASCIZ  'BIT5'
                638 003612      102     111     124  12$:    .ASCIZ  'BIT4'
                639 003617      102     111     124  13$:    .ASCIZ  'BIT3'
                640 003624      102     111     124  14$:    .ASCIZ  'BIT2'
                641 003631      102     111     124  15$:    .ASCIZ  'BIT1'
                642 003636      102     111     124  16$:    .ASCIZ  'BIT0'
                643                                          .EVEN
                644 003644      124     123     123  SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
                645 003677      124     123     123  SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
                646 003732      040     040     116  NXR:    .ASCIZ  /  NON-EXISTANT DEVICE REGISTER/
                647 003771      045     101     040  NXRX:   .ASCIZ  /%A  ADDRESS: %O6/
                648 004012      045     101     040  TSSX:   .ASCII  /%A  TSBA,TSSR EXP'D: %O6%A,%O6%N/
                649 004052      045     101     040          .ASCIZ  /%A  TSBA,TSSR REC'D: %O6%A,%O6/
                650 004111      045     116     045  FUSI:   .ASCII  /%N%A/
                651 004115      040     040     125  USI:    .ASCIZ  /  UNEXPECTED INTERRUPT/
                652 004144      040     040     111  NSI:    .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
                653 004207      045     116     045  FNOINTR:        .ASCII  /%N%A/
                654 004213      040     040     116  NOINTR: .ASCIZ  /  NO INTERRUPT WAS GENERATED/
                655 004250      040     040     111  IFAULT: .ASCIZ  /  INTERRUPT FAULT/
                656 004272      045     101     040  INTX:   .ASCIZ  /%A  CPU PC: %O6%A  TSBA: %O6/
                657 004327      040     040     042  NOINIT: .ASCIZ  /  "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
                658 004401      040     040     042  NSINIT: .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE CPU/
                659 004451      040     040     042  BRINIT: .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE CPU/
                660
                661 004521      000                  NUL:    .ASCIZ  //
                662 004522      045     116     000  NULCR:  .ASCIZ  /%N/
                663 004525      045     101     040  EXPGOT: .ASCIZ  /%A EXP'D: %O6%A, REC'D: %O6/
                664 004561      045     116     045  EXPGT2: .ASCIZ  /%N%A EXP'D: %O6%A, %O6%N%A REC'D: %O%A, %O6/
                665 004635      045     101     040  DUAD12: .ASCIZ  /%A REG(W) WRITTEN TO: %O6%A REG(R) READ; EXP'D: %O6%A, REC'D: %O6/
                666 004737      123     101     115  PKTRAM::        .ASCIZ  'RAM Contents Do Not Match Packet Sent'
                667 005005      040     040     103  SCME:   .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
                668 005050      127     122     111  WRTMSG: .ASCIZ  'WRITE CHARACTERISTICS Failed'
                669 005105      124     123     123  WRTERR: .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
                670 005200      124     123     123  RDERR:  .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
                671 005272      106     101     124  SCHERR: .ASCIZ  'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
                672 005364      105     122     122  RETERR: .ASCIZ  'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
                673 005452      045     116     045  NOMEM:  .ASCIZ  '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT, ******%N'
                674                                          .EVEN
                675
                676                                          .SBTTL  GLOBAL ERROR REPORT SECTION
                677
                678                                  ;++
                679                                  ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
                680                                  ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
                681                                  ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
                682                                  ;--
                683
                684 005546                                       BGNMSG  NXRERR          ;NON-EXISTANT DEVICE REGISTER.
                    005546                               NXRERR::
                685 005546                                       PRINTX  #NXRX,NODEV     ;NODEV = NEXM ADDRESS.
                    005546  013746  003112'                       MOV     NODEV,-(SP)
                    005552  012746  003771'                       MOV     #NXRX,-(SP)
                    005556  012746  000002                       MOV     #2,-(SP)
```

```
     005562  010600                          MOV     SP,R0
     005564  104415                          TRAP    C$PNTX
     005566  062706  000006                  ADD     #6,SP
686  005572  004737  005600'                JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
687  005576                                  ENDMSG
     005576                          L10002:
     005576  104423                          TRAP    C$MSG
688
689
690                                  ;
691                                  ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
692                                  ; TO ANY OF THE ABOVE ERROR SIGNATURES.
693                                  ;
694  005600  005727                  EXTEND: TST     (PC)+
695  005602  000000                  EXTA:   0                       ; 0 = NO EXTENSION.
696  005604  001402                          BEQ     1$
697  005606  004777  177770                  JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
698  005612                          1$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
     005612  012746  004522'                MOV     #NULCR,-(SP)
     005616  012746  000001                  MOV     #1,-(SP)
     005622  010600                          MOV     SP,R0
     005624  104415                          TRAP    C$PNTX
     005626  062706  000004                  ADD     #4,SP
699  005632  000207                          RTS     PC
700
701                                          .SBTTL  PRITSSR - PRINT TSSR CONTENTS
702
703                                  ;+
704                                  ;
705                                  ;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
706                                  ;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
707                                  ;BY A MESSAGE PRINTING ROUTINE
708                                  ;
709                                  ;INPUTS:
710                                  ;
711                                  ;       R1      CONTENTS OF TSSR
712                                  ;
713                                  ;SUBORDINATE ROUTINES:
714                                  ;
715                                  ;       CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
716                                  ;
717                                  ;-
718
719  005634                          PRITSSR:
720  005634                                  SAVREG                  ;SAVE GENERAL REGISTERS
721  005640  010104                          MOV     R1,R4           ;SAVE THE TSSR CONTENTS
722  005642                                  PRINTB  #TSSRFOR,R4     ;PRINT THE CONTENTS OF TSSR
     005642  010446                          MOV     R4,-(SP)
     005644  012746  006225'                MOV     #TSSRFOR,-(SP)
     005650  012746  000002                  MOV     #2,-(SP)
     005654  010600                          MOV     SP,R0
     005656  104414                          TRAP    C$PNTB
     005660  062706  000006                  ADD     #6,SP
723  005664  010400                          MOV     R4,R0           ;GET TSSR BACK FOR CHKAMB
724  005666  004737  015654'                JSR     PC,CHKAMB       ;ARE CONTENTS AMBIGUOUS ?
725  005672  103410                          BCS     5$              ;BRANCH IF NOT
726  005674                                  PRINTX  #AMBTSSR        ;SHOW CONTENTS ARE AMBIGUOUS
```

```
        005674  012746  006445'                MOV     #AMBTSSR,-(SP)
        005700  012746  000001                 MOV     #1,-(SP)
        005704  010600                          MOV     SP,R0
        005706  104415                          TRAP    C$PNTX
        005710  062706  000004                 ADD     #4,SP
    727 005714  010403                  5$:     MOV     R4,R3           ;CONTENTS OF TSSR
    728 005716  042703  001476                  BIC     #HIADDR!FATERR!TERCLS,R3        ;CLEAR ALL MULTIPLE BIT FIELDS
    729 005722  001434                          BEQ     20$             ;NO BITS ARE SET
    730 005724  012702  002630'                 MOV     #TMPBFR,R2      ;TEMPORARY ASCII BUFFER
    731 005730  012701  003474'                 MOV     #TSSRBIT,R1     ;ASCII EQUIVALENT OF BITS
    732 005734  005703                  10$:    TST     R3              ;REMAINING BITS TO CONVERT
    733 005736  001413                          BEQ     15$             ;BRANCH WHEN ALL ARE DONE
    734 005740  000241                          CLC                     ;CLEAR CARRY FOR SHIFT
    735 005742  006103                          ROL     R3              ;SHIFT NEXT BIT TO CARRY
    736 005744  103006                          BCC     13$             ;BRANCH IF BIT NOT SET
    737 005746  011100                          MOV     (R1),R0         ;POINTER TO BIT DEFINITION
    738 005750  112022                  11$:    MOVB    (R0)+,(R2)+     ;MOVE ASCIZ TO BUFFER
    739 005752  001376                          BNE     11$             ;MOVE ALL BITS
    740 005754  112762  000054  177777          MOVB    #',,-1(R2)      ;INSERT A COMMA TO TERMINATE
    741 005762  005721                  13$:    TST     (R1)+           ;POINT TO NEXT DESCRIPTION
    742 005764  000763                          BR      10$             ;GET THE REMAINING BITS
    743 005766  105042                  15$:    CLRB    -(R2)           ;TERMINATE THE LINE
    744 005770                                  PRINTX  #TSSDEF,#TMPBFR ;PRINT THE BIT DEFINITIONS
        005770  012746  002630'                 MOV     #TMPBFR,-(SP)
        005774  012746  006416'                 MOV     #TSSDEF,-(SP)
        006000  012746  000002                  MOV     #2,-(SP)
        006004  010600                          MOV     SP,R0
        006006  104415                          TRAP    C$PNTX
        006010  062706  000006                  ADD     #6,SP
    745
    746 006014  010403                  20$:    MOV     R4,R3           ;GET THE TSSR CONTENTS
    747 006016  042703  177761                  BIC     #^TCTERCLS,R3   ;CLEAR ALL BUT TERMINATION
    748 006022  016303  006506'                 MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
    749 006026                                  PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
        006026  010346                          MOV     R3,-(SP)
        006030  012746  006306'                 MOV     #TCOASC,-(SP)
        006034  012746  000002                  MOV     #2,-(SP)
        006040  010600                          MOV     SP,R0
        006042  104415                          TRAP    C$PNTX
        006044  062706  000006                  ADD     #6,SP
    750 006050  010403                          MOV     R4,R3           ;TSSR CONTENTS AGAIN
    751 006052  042703  177717                  BIC     #^TCFATERR,R3   ;CLEAR ALL BUT FATAL TERMINATION
    752 006056  001416                          BEQ     25$             ;DON'T PRINT IF ZERO
    753 006060  006203                          ASR     R3
    754 006062  006203                          ASR     R3
    755 006064  006203                          ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
    756 006066  016303  007046'                 MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
    757 006072                                  PRINTX  #TFCASC,R3      ;PRINT THE FATAL TERMINATION CODE
        006072  010346                          MOV     R3,-(SP)
        006074  012746  006347'                 MOV     #TFCASC,-(SP)
        006100  012746  000002                  MOV     #2,-(SP)
        006104  010600                          MOV     SP,R0
        006106  104415                          TRAP    C$PNTX
        006110  062706  000006                  ADD     #6,SP
    758 006114  042704  176377          25$:    BIC     #^TCHIADDR,R4   ;CLEAR ALL BUT EXTENDED ADDRESS
    759 006120  001411                          BEQ     30$             ;DON'T PRINT IF ZERO
    760 006122                                  PRINTX  #TEXASC,R4      ;PRINT THE EXTENDED ADDRESS BITS
```

```
        006122  010446                          MOV     R4,-(SP)
        006124  012746  006245'                  MOV     #TEXASC,-(SP)
        006130  012746  000002                  MOV     #2,-(SP)
        006134  010600                          MOV     SP,R0
        006136  104415                          TRAP    C$PNTX
        006140  062706  000006                  ADD     #6,SP
    761 006144  013703  002176'         30$:    MOV     EPRTSW,R3               ;PRINT MEAASGE BUFFER ADDRESS
    762 006150                                  PRINTX  R3                      ;PRINT PROPER MESSAGE
        006150  010346                          MOV     R3,-(SP)
        006152  012746  000001                  MOV     #1,-(SP)
        006156  010600                          MOV     SP,R0
        006160  104415                          TRAP    C$PNTX
        006162  062706  000004                  ADD     #4,SP
    763 006166  000207                          RTS     PC              ;RETURN TO CALLER
    764
    766 006170                          EPRT2:
    767 006170      045     116     045 EPRT1:  .ASCIZ   '%N%A *****REPLACE M7455*****'
    782 006225      045     116     045 TSSRFOR:        .ASCIZ  '%N%A TSSR = %O6'
    783 006245      045     116     045 TEXASC:         .ASCIZ  '%N%A Extended Address Bits = %O6'
    784 006306      045     116     045 TCOASC:         .ASCIZ  '%N%A Termination Class Code = %T'
    785 006347      045     116     045 TFCASC:         .ASCIZ  '%N%A Fatal Termination Class Code = %T'
    786 006416      045     116     045 TSSDEF:         .ASCIZ  '%N%A TSSR Bits Set: %T'
    787 006445      045     116     045 AMBTSSR:        .ASCIZ  '%N%A TSSR Contents Are Ambiguous'
    788                                 .EVEN
    789 006506  006526' 006551' 006577' TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
    790 006526      116     157     162 1$:     .ASCIZ  'Normal Termination'
    791 006551      124     145     162 2$:     .ASCIZ  'Termination Condition'
    792 006577      124     141     160 3$:     .ASCIZ  'Tape Status Alert'
    793 006621      106     165     156 4$:     .ASCIZ  'Function Reject'
    794 006641      122     145     143 5$:     .ASCIZ  'Recoverable Error - Tape Position One Record Down'
    795 006723      122     145     143 6$:     .ASCIZ  'Recoverable Error - Tape Was Not Moved'
    796 006772      125     156     162 7$:     .ASCIZ  'Unrecoverable Error'
    797 007016      106     141     164 8$:     .ASCIZ  'Fatal Controller Error'
    798                                 .EVEN
    799
    800 007046  007056' 007112' 007123' TSFCOD: .WORD   1$,2$,3$,4$
    801 007056      111     156     164 1$:     .ASCIZ  'Internal Diagnostic Failure'
    802 007112      122     145     163 2$:     .ASCIZ  'Reserved'
    803 007123      102     165     163 3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
    804 007167      122     145     163 4$:     .ASCIZ  'Reserved'
    805                                 .EVEN
    806
    807                                 .SBTTL  PRIPKT  - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
    808
    809                         ;+
    810                         ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
    811                         ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
    812                         ;
    813                         ;INPUT:
    814                         ;
    815                         ;       R0      NUMBER OF WORDS IN PACKET
    816                         ;       R3      HIGH ORDER COMMAND PACKET ADDRESS
    817                         ;       R4      ADDRESS OF COMMAND PACKET
    818                         ;
    819                         ;       NOTE:   R3 IS IGNORLD IF THE KTENABLE FLAG IS CLEAR.
    820                         ;-
    821
```

```
                        PRIPKT::
007200                          SAVREG                  ;SAVE THE REGISTERS
007200  010005                  MOV     R0,R5           ;SAVE NO. OF WORDS IN PACKET
007204  005737  003132'          TST     KTENABLE        ;ABOVE 28K UNDER TEST?
007210  001001                  BNE     10$             ;BR IF YES
007212  005003                  CLR     R3              ;SET HIGH ORDER ADDRESS TO 0
007214  010301          10$:    MOV     R3,R1           ;COPY HIGH ORDER ADDRESS
007216  010400                  MOV     R4,R0           ;GET LOWER ADDRESS
007220  006100                  ROL     R0              ;SHIFT BIT 15 INTO C BIT
007222  006101                  ROL     R1              ;AND INTO HIGH ORDER.
007224                          PRINTB  #PKTADD,R1,R4   ;PRINT PACKET ADDRESS
007226  010446                  MOV     R4,-(SP)
007230  010146                  MOV     R1,-(SP)
007232  012746  007364'          MOV     #PKTADD,-(SP)
007236  012746  000003          MOV     #3,-(SP)
007242  010600                  MOV     SP,R0
007244  104414                  TRAP    C$PNTB
007246  062706  000010          ADD     #10,SP
007252  010300          15$:    MOV     R3,R0           ;GET HIGH ORDER ADDRESS
007254  001404                  BEQ     20$             ;BR IF NOT ABOVE 28K.
007256  010401                  MOV     R4,R1           ;GET LOW ORDER ADDRESS
007260  004737  017130'          JSR     PC,SETMAP       ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
007264  010004                  MOV     R0,R4           ;GET RETURNED PAR6 ADDRESS BIAS
007266  005001          20$:    CLR     R1              ;SAVE WORD NUMBER
007270  012402          25$:    MOV     (R4)+,R2        ;GET PACKET CONTENTS
007272                          PRINTB  #PKTFRM,R1,R2   ;PRINT THE DATA
007272  010246                  MOV     R2,-(SP)
007274  010146                  MOV     R1,-(SP)
007276  012746  007326'          MOV     #PKTFRM,-(SP)
007302  012746  000003          MOV     #3,-(SP)
007306  010600                  MOV     SP,R0
007310  104414                  TRAP    C$PNTB
007312  062706  000010          ADD     #10,SP
007316  005201                  INC     R1              ;NEXT WORD NUMBER
007320  020105                  CMP     R1,R5           ;DONE ALL PACKET WORDS?
007322  002762                  BLT     25$             ;LOOP TILL ALL DONE
007324  000207                  RTS     PC              ;RETURN

007326  045  116  045  PKTFRM: .ASCIZ  '%N%A Packet Word #%D1%A = %O6'
007364  045  116  045  PKTADD: .ASCIZ  '%N%A Packet Address = %O1%O5'
                        .EVEN


                        .SBTTL  PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

                ;+
                ;
                ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
                ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
                ;
                ;INPUTS:
                ;
                ;       R1      RECEIVED DATA
                ;       R2      EXPECTED DATA
                ;
                ;OUTPUT:
                ;
```

```
                                ;       R0      XOR OF EXPECTED/RECEIVED DATA
                                ;
                                ;-

007422                          PRIBXOR::
007422                                  SAVREG                          ;SAVE THE REGISTERS
007426  010203                          MOV     R2,R3                   ;EXPECTED DATA
007430                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007440  012700  177400                  MOV     #^C<377>,R0             ;BYTE MASK
007444  040001                          BIC     R0,R1                   ;SAVE LOW BYTE RECV
007446  040002                          BIC     R0,R2                   ;SAVE LOW BYTE EXPD
007450  040003                          BIC     R0,R3                   ;SAVE LOW BYTE XOR
007452                                  PRINTB  #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
007452  010346                          MOV     R3,-(SP)
007454  010146                          MOV     R1,-(SP)
007456  010246                          MOV     R2,-(SP)
007460  012746  007504'                 MOV     #XORBFOR,-(SP)
007464  012746  000004                  MOV     #4,-(SP)
007470  010600                          MOV     SP,R0
007472  104414                          TRAP    C^PNTB
007474  062706  000012                  ADD     #12,SP
007500  010300                          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007502  000207                          RTS     PC                      ;RETURN TO CALLER

007504      045     116     045 XORBFOR:        .ASCIZ  '#N#A EXPD: #03#A RECV: #03#A XOR: #03'
                                                .EVEN


                                        .SBTTL  PRIXOR  - PRINT EXPD, RECV AND XOR

                                ;+
                                ;
                                ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      XOR OF EXPECTED/RECEIVED DATA
                                ;
                                ;-

007552                          PRIXOR::
007552                                  SAVREG                          ;SAVE THE REGISTERS
007556  010203                          MOV     R2,R3                   ;EXPECTED DATA
007560                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007570                                  PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
007570  010346                          MOV     R3,-(SP)
007572  010146                          MOV     R1,-(SP)
007574  010246                          MOV     R2,-(SP)
007576  012746  007622'                 MOV     #XORFOR,-(SP)
007602  012746  000004                  MOV     #4,-(SP)
007606  010600                          MOV     SP,R0
```

```
                        ;
                        ;PRINT MEMORY ADDRESS
                        ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                        ;
                        ; IMPLICIT INPUTS
                        ;
                        ;       ERRHI   - HIGH ORDER ADDRESS
                        ;       ERRLO   - LOW ORDER ADDRESS
                        ;
                        ;-
007770                  PRIADD:
007770                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
007774  013700  002234'          MOV     ERRHI,R0                ;GET HIGH ADDRESSS
010000  013701  002236'          MOV     ERRLO,R1                ;GET LOW ADDRESS
010004  010102                  MOV     R1,R2                   ;COPY LOW ADDRESS
010006  006101                  ROL     R1                      ;SHIFT BIT 15 TO C BIT
010010  006100                  ROL     R0                      ;SHIFT INTO HIGH ORDER
010012                          PRINTB  #PRIA0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
010012  010246                  MOV     R2,-(SP)
010014  010046                  MOV     R0,-(SP)
010016  012746  010040'          MOV     #PRIA0,-(SP)
010022  012746  000003          MOV     #3,-(SP)
010026  010600                  MOV     SP,R0
010030  104414                  TRAP    C$PNTB
010032  062706  000010          ADD     #10,SP
010036  000207                  RTS     PC                      ;RETURN

010040  045  116  045  PRIA0:   .ASCIZ  '%N%A MEMORY ERROR ADDRESS = %01%05'
                        .EVEN


                        .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
                        ;+
                        ;
                        ;PRINT MEMORY ADDRESS
                        ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                        ;
                        ; IMPLICIT INPUTS
                        ;
                        ;       ERRHI   - HIGH ORDER ADDRESS
                        ;       ERRLO   - LOW ORDER ADDRESS
                        ;
                        ;-
010104                  PRITADD:
010104                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010110  013702  002234'          MOV     ERRHI,R2                ;GET HIGH ADDRESSS
010114  013701  002236'          MOV     ERRLO,R1                ;GET LOW ADDRESS
                                ;MOV     R1,R2                   ;COPY LOW ADDRESS
                                ;ROL     R1                      ;SHIFT BIT 15 TO C BIT
                                ;ROL     R0                      ;SHIFT INTO HIGH ORDER
010120                          PRINTB  #PRIT0,R1               ;PRINT MEMORY ADDRESS LOW IN ERROR
010120  010146                  MOV     R1,-(SP)
010122  012746  010166'          MOV     #PRIT0,-(SP)
010126  012746  000002          MOV     #2,-(SP)
010132  010600                  MOV     SP,R0
010134  104414                  TRAP    C$PNTB
010136  062706  000006          ADD     #6,SP
```

```
1001 010142                                          PRINTB  #PRIT1,R2       ;PRINT MEMORY ADDRESS HIGH IN ERROR
     010142  010246                                  MOV     R2,-(SP)
     010144  012746  010231'                         MOV     #PRIT1,-(SP)
     010150  012746  000002                          MOV     #2,-(SP)
     010154  010600                                  MOV     SP,R0
     010156  104414                                  TRAP    C$PNTB
     010160  062706  000006                          ADD     #6,SP
1002 010164  000207                                  RTS     PC              ;RETURN
1003
1004 010166     045     116     045  PRIT0:  .ASCIZ  '%N%A MEMORY TEST ADDRESS LOW = %O5'
1005 010231     045     116     045  PRIT1:  .ASCIZ  '%N%A MEMORY TEST ADDRESS HIGH = %O6'
1006                                         .EVEN
1007
1008
1009                                         .SBTTL  SPACE   - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
1010
1011                                 ;+
1012                                 ;
1013                                 ;ROUTINE TO ISSUE A SPACE RECORDS
1014                                 ;COMMAND (FORWARD OR REVERSE)
1015                                 ;
1016                                 ;INPUT:
1017                                 ;
1018                                 ;       R3      NUMBER OF RECORDS TO BE SPACED OVER
1019                                 ;               BIT15 CONTROLS DIRECTION
1020                                 ;               BIT15 = 0 IS FORWARD
1021                                 ;               BIT15 = 1 IS REVERSE
1022                                 ;       R5      FIRST DEVICE UNIBUS ADDRESS
1023                                 ;
1024                                 ;       REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
1025                                 ;
1026                                 ;OUTPUT:
1027                                 ;
1028                                 ;       CARRY   SET - SPACE RECORDS COMMAND OK
1029                                 ;               CLR - SPACE RECORDS FAILED
1030                                 ;
1031                                 ;
1032                                 ;       R0      THE CONTENTS OF R4 IS MOVED TO R0
1033                                 ;
1034                                 ;
1035                                 ;IMPLICIT OUTPUT:
1036                                 ;
1037                                 ;       TAPE HAS BEEN MOVED
1038                                 ;
1039                                 ;SIDE EFFECTS:
1040                                 ;
1041                                 ;
1042                                 ;-
1043
1044 010276                          SPACE::
1045 010276                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
1046 010302  012737  000764  010470'         MOV     #500.,SDELAY            ;SET UP DELAY
1047 010310  012737  140010  010460'         MOV     #140010,80$             ;SET UP COMMAND, SPACE FORWARD
1048 010316  005703                          TST     R3                      ;CHECK FOR DIRECTION
1049 010320  100403                          BMI     5$                      ;BR, IF REVERSE INDICATED
1050 010322  010337  010462'                 MOV     R3,90$                  ;LOAD UP NUMBER OF RECORDS TO SPACE
1051 010326  000407                          BR      10$                     ;GO DO COMMAND
```

```
1052 010330  042703  100000          5$:     BIC     #BIT15,R3               ;CLEAR DIRECTION BIT
1053 010334  010337  010462'                 MOV     R3,90$                  ;LOAD UP NUMBER OF RECORDS TO SPACE
1054 010340  052737  000400  010460'         BIS     #BIT8,80$               ;SET REVERSE BIT IN COMMAND PACKET
1055 010346  012704  010460'          10$:    MOV     #80$,R4                 ;SET UP R4 WITH PACKET ADDRESS
1056 010352  010465  000000                   MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
1057 010356  004737  016060'          15$:    JSR     PC,WAITF                ;WAIT FOR SSR
1058 010362  103420                           BCS     20$                     ;BR, IF SSR IS SET AND OK
1059 010364                                   DELAY   250             ;DELAY ABOUT .25 SECONDS
     010364  012727  000250                   MOV     #250,(PC)+
     010370  000000                           .WORD   0
     010372  013727  002116'                   MOV     L$DLY,(PC)+
     010376  000000                           .WORD   0
     010400  005367  177772                   DEC     -6(PC)
     010404  001375                           BNE     .-4
     010406  005367  177756                   DEC     -22(PC)
     010412  001367                           BNE     .-20
1060 010414  005337  010470'                 DEC     $DELAY                  ;BUMP DELAY COUNTER DOWN
1061 010420  001356                           BNE     15$                     ;BR, IF MORE DELAY
1062 010422  000411                           BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
1063 010424  016501  000002          20$:    MOV     TSSR(R5),R1             ;READ TSSR
1064 010430  012702  000200                   MOV     #SSR,R2                 ;SET UP EXPECTED
1065 010434  020201                   25$:    CMP     R2,R1                   ;ARE THEY OK
1066 010436  001401                           BEQ     40$                     ;BR, IF EQUAL = OK
1067 010440  000402                           BR      60$                     ;TROUBLE EXIT
1068 010442  000261                   40$:    SEC                             ;SET CARRY NO TROUBLE
1069 010444  000401                           BR      70$                     ;EXIT
1070 010446  000241                   60$:    CLC                             ;CARRY CLEAR = ERROR
1071 010450                            70$:
1072 010450  010400                           MOV     R4,R0                   ;PASS PACKET ADDRESS
1073 010452  000207                           RTS     PC                      ;RETURN
1074
1075
1076
1077
1078
1079
1081 010454                                   .BLKB   10-<.-TSV2&7>
1083
1084
1085 010460  000000           80$:    .WORD
1086                                   ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1087 010462  000000           90$:    .WORD
1088 010464  000000                   .WORD
1089 010466  000000                   .WORD
1090 010470  000000           $DELAY: .WORD   0                               ;DELAY COUNTER
1091                                   .EVEN
1092
1093
1094                                   .SBTTL  WRTCHR  - WRITE CHARACTERISTICS COMMAND
1095
1096                                   ;+
1097                                   ;
1098                                   ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1099                                   ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1100                                   ;
1101                                   ;INPUT:
1102                                   ;
```

```
                        ;       R4      ADDRESS OF PACKET FROM TEST
                        ;       R5      FIRST DEVICE UNIBUS ADDRESS
                        ;       REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
                        ;
                        ;OUTPUT:
                        ;
                        ;       R0      TSSR CONTENTS
                        ;       CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
                        ;               CLR - WRITE CHARACTERISTICS FAILED
                        ;
                        ;IMPLICIT OUTPUT:
                        ;
                        ;       MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
                        ;       SOFTWARE SWITCHES SET AS FOLDOWS:
                        ;               EXTFEA = EXTENDED FEATURES PRESENT
                        ;               BENBSW = BUFFER ENABLE SWITCH ON OR OFF
                        ;
                        ;
                        ;SIDE EFFECTS:
                        ;
                        ;
                        ;-

010472                  WRTCHR::
010472                          SAVREG                          ;SAVE THE GENERAL REGISTERS
010476  005037  002226'          CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010502  005[illegible]  02224'          CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SW SWITCH
010506  01[illegible]  00000    10$:    MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010512  0[illegible]  16146'            JSR     PC,CHKTSSR              ;WAIT FOR SSR
010516  1[illegible]                  BCS     20$                     ;BR, IF SSR IS SET AND OK
010520  0[illegible]                  BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
010522  016501  000002        20$:    MOV     TSSR(R5),R1             ;READ TSSR
010526  012702  000200                MOV     #SSR,R2                 ;SET UP EXPECTED
010532  032701  000100                BIT     #OFL,R1                 ;WAS OFF LINE SET IN TSSR
010536  001402                        BEQ     25$                     ;BR, IF NO OFL SET
010540  052702  000100                BIS     #OFL,R2                 ;MAKE THEM LOOK ALIKE
010544  020201          25$:    CMP     R2,R1                   ;ARE THEY OK
010546  001401                        BEQ     40$                     ;BR, IF EQUAL = OK
010550  000421                        BR      60$                     ;TROUBLE EXIT
010552  062704  000010        40$:    ADD     #8.,R4                  ;POINT TO WRT CHARA DATA PACKET
010556  011403                        MOV     (R4),R3                 ;GET ADDRESS OF MESSAGE BUFFER
010560  032763  000200  000012        BIT     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURES BIT SET?
010566  001402                        BEQ     45$                     ;BR IF NO
010570  005237  002224'              INC     EXTFEA                  ;SET EXTENDED FEATURES SW SWITCH
010574                          45$:
010574  032763  000100  000012        BIT     #X2.BUFE,XST2(R3)                               ;BUFFER ENABLE SWITCH SET
010602  001402                        BEQ     50$                     ;BR, IF SWITCH NOT SET
010604  005237  002226'              INC     BENBSW                  ;SET SOFTWARE SWITCH FOR ENABLED
010610                          50$:
010610  000261                        SEC                             ;SET CARRY NO TROUBLE
010612  000401                        BR      70$                     ;EXIT
010614  000241          60$:    CLC                             ;CARRY CLEAR = ERROR
010616  016500  000002        70$:    MOV     TSSR(R5),R0             ;RETURN TSSR CONTENTS
010622  000207                        RTS     PC                      ;RETURN


                        .SBTTL  REWIND  - POSITION TAPE (REWIND) COMMAND
```

```
1160
1161                            ;*
1162                            ;
1163                            ;THIS ROUTINE WILL REWIND THE SELECTED TAPE.
1164                            ;
1165                            ;       CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
1166                            ;                TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
1167                            ;                SSR TO SET IN THE TSSR
1168                            ;
1169                            ;
1170                            ;CALLING SEQUENCE:
1171                            ;
1172                            ;       DO A SOFT INIT
1173                            ;       DO A WRITE CHARACTERISTICS
1174                            ;       JSR     PC,REWIND
1175                            ;
1176                            ;INPUT:
1177                            ;
1178                            ;       R5      FIRST DEVICE UNIBUS ADDRESS
1179                            ;
1180                            ;
1181                            ;OUTPUT
1182                            ;
1183                            ;       R0      THE CONTENTS OF R4 IS PASSED TO R0
1184                            ;
1185                            ;
1186                            ;-
1187 010624                    REWIND::
1188 010624                            SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
1189 010630  012704  010720'          MOV     #RWPACK,R4              ;GET PACKET ADDRESS
1190 010634  010465  000000           MOV     R4,TSDB(R5)             ;SEND PACKET ADDRESS TO EXECUTE
1191 010640  012703  000550           MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
1192 010644  004737  016060'  10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1193 010650  103417                   BCS     20$                     ;LEAVE WHEN SSR IS SET
1194 010652                           DELAY   250.                    ;WAIT FOR .25 SECONDS
     010652  012727  000372           MOV     #250.,(PC)+
     010656  000000                   .WORD   0
     010660  013727  002116'          MOV     L$DLY,(PC)+
     010664  000000                   .WORD   0
     010666  005367  177772           DEC     -6(PC)
     010672  001375                   BNE     .-4
     010674  005367  177756           DEC     -22(PC)
     010700  001367                   BNE     .-20
1195 010702  005303                   DEC     R3                      ;BUMP COUNTER DOWN
1196 010704  001357                   BNE     10$                     ;KEEP GOING
1197 010706  000241                   CLC                             ;CLEAR CARRY TO SET ERROR
1198 010710  010400           20$:    MOV     R4,R0                   ;PASS THE PACKET ADDRESS
1199 010712  000207                   RTS     PC                      ;RETURN
1200
1201
1203 010714                           .BLKB   10-<.-TSV2&7>
1205 010720                    RWPACK:
1206 010720  102010                   .WORD   102010                  ;POSTION COMMAND (REWIND)
1207 010722  000000                   .WORD   0                       ;NOT USED
1208
1209
1210                                   .SBTTL  CKRAM   - COMPARE RAM TO I/O PACKET
```

```
1211
1212                                    ;+
1213                                    ;
1214                                    ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
1215                                    ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1216                                    ;
1217                                    ;INPUT:
1218                                    ;
1219                                    ;       R4      ADDRESS OF   F COMMAND PACKET
1220                                    ;       R5      FIRST DEVICE UNIBUS ADDRESS
1221                                    ;
1222                                    ;OUTPUT:
1223                                    ;
1224                                    ;       CARRY   SET - RAM MATCHES PACKET
1225                                    ;               CLR - RAM DOES NOT MATCH PACKET
1226                                    ;
1227                                    ;IMPLICIT OUTPUT:
1228                                    ;
1229                                    ;       THE TABLE RAMDATA IS FILLED WITH THE
1230                                    ;       DATA HELD IN RAM.
1231                                    ;       RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
1232                                    ;
1233                                    ;SIDE EFFECTS:
1234                                    ;
1235                                    ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1236                                    ;
1237                                    ;-
1238
1239 010724                            CKRAM::
1240 010724                                    SAVREG                          ;SAVE THE GENERAL REGISTERS
1241 010730  012701  002240'                   MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
1242 010734  012702  000201                    MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
1243 010740  005003                            CLR     R3                      ;CLEAR THE ERROR FLAG
1244 010742  004737  016146'                   JSR     PC,CHKTSSR              ;WAIT FOR SSR
1245 010746  112765  000000  000000            MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
1246 010754  004737  016146'           10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1247 010760  010265  000000                    MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
1248 010764  004737  016146'                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1249 010770  116511  000000                    MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
1250 010774  122124                            CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
1251 010776  001401                            BEQ     20$                     ;BRANCH IF OK
1252 011000  005203                            INC     R3                      ;SET ERROR FLAG
1253 011002  005202                    20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
1254 011004  020227  000210                    CMP     R2,#RMPKTEND            ;REACHED END YET ?
1255 011010  003761                            BLE     10$                     ;BRANCH TILL ALL READ
1256 011012  005703                            TST     R3                      ;WAS AN ERROR FOUND ?
1257 011014  001402                            BEQ     30$                     ;BRANCH IF NOT
1258 011016  000241                            CLC                             ;CLEAR CARRY TO SHOW ERROR
1259 011020  000401                            BR      50$                     ;AND EXIT
1260 011022  000261                    30$:    SEC                             ;SHOW GOOD COMPARE
1261 011024  012737  000010  002300'   50$:    MOV     #8.,RAMSIZ              ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
1262 011032  000207                            RTS     PC                      ;RETURN
1263
1264
1265                                           .SBTTL  CKRAM2  - COMPARE RAM TO I/O CHARACTERISTICS DATA
1266                                    ;+
1267                                    ;
```

```
                        ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
                        ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
                        ;
                        ;INPUT:
                        ;
                        ;       R4      ADDRESS OF THE CHARACTERISTICS DATA
                        ;       R5      FIRST DEVICE UNIBUS ADDRESS
                        ;
                        ;OUTPUT:
                        ;
                        ;       CARRY   SET - RAM MATCHES PACKET
                        ;               CLR - RAM DOES NOT MATCH PACKET
                        ;
                        ;IMPLICIT OUTPUT:
                        ;
                        ;       THE TABLE RAMDATA IS FILLED WITH THE
                        ;       DATA HELD IN RAM.
                        ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
                        ;
                        ;SIDE EFFECTS:
                        ;
                        ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
                        ;
                        ;-

011034                  CKRAM2::
011034                          SAVREG                          ;SAVE THE GENERAL REGISTERS
011040  012701  002240'          MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011044  012702  000167          MOV     #RMCHBEG,R2             ;BYTE ADDRESS OF FIRST RAM DATA
011050  005003                  CLR     R3                      ;CLEAR THE ERROR FLAG
011052  004737  016146'          JSR     PC,CHKTSSR              ;WAIT FOR SSR
011056  112765  000000  000000    MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011064  004737  016146'  10$:     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011070  010265  000000          MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011074  004737  016146'          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011100  116511  000000          MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011104  122124                  CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011106  001401                  BEQ     20$                     ;BRANCH IF OK
011110  005203                  INC     R3                      ;SET ERROR FLAG
011112  005202          20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011114  012737  000010  002300' MOV     #8.,RAMSIZ              ;ASSUME EXTFEA NOT SET
011122  005737  002224'          TST     EXTFEA                  ;IS THE SOFTWARE EXTENDED FEATURES SET
011126  001407                  BEQ     25$                     ;BR, IF NOT SET
011130  012737  000012  002300' MOV     #10.,RAMSIZ             ;SET RAMSIZ FOR EXTEND FEATURES
011136  020227  000200          CMP     R2,#RMCHEND             ;AT END OF EXTENDED BUFFER
011142  003750                  BLE     10$                     ;BR, IF NOT AT END YET
011144  000403                  BR      27$                     ;AT END BRANCH
011146  020227  000176  25$:    CMP     R2,#RMCHEND-2           ;REACHED END YET ?
011152  003744                  BLE     10$                     ;BRANCH TILL ALL READ
011154  005703          27$:    TST     R3                      ;WAS AN ERROR FOUND ?
011156  001402                  BEQ     30$                     ;BRANCH IF NOT
011160  000241                  CLC                             ;CLEAR CARRY TO SHOW ERROR
011162  000401                  BR      50$                     ;AND EXIT
011164  000261          30$:    SEC                             ;SHOW GOOD COMPARE
011166  000207          50$:    RTS     PC                      ;RETURN
```

```
                        .SBTTL  CKMSG   - COMPARE WRITE CHAR. MESSAGE BUFFERS
                ;+
                ;
                ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                ;ERROR PRINT ROUTINES.
                ;
                ;INPUT:
                ;
                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
                ;       R2      EXPD MESSAGE BUFFER ADDRESS
                ;OUTPUT:
                ;
                ;       CARRY   SET - MESSAGE BUFFERS MATCH
                ;               CLR -MESSAGE BUFFERS DON'T MATCH
                ;
                ;IMPLICIT OUTPUT:
                ;
                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
                ;       RECMSG          BUFFER IS SET TC RECV DATA
                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                ;
                ;-
011170          CKMSG::
011170                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011174  010037  002302'         MOV     R0,RCVHIADD     ;SAVE RECV HIGH ADDRESS
011200  010137  002304'         MOV     R1,RCVLOAD      ;SAVE RECV LOW ADDRESS
011204  005737  003132'         TST     KTENABLE        ;TESTING ABOVE 28K?
011210  001403                  BEQ     10$             ;BR IF NO
011212  004737  017130'         JSR     PC,SETMAP       ;RETURN ADDRESS BIASED TO PAR6 IN R0
011216  010001                  MOV     R0,R1           ;GET RETURNED ADDRESS BIASED TO PAR6
011220  005004          10$:    CLR     R4              ;WORD IN BUFFER
011222  005003                  CLR     R3              ;CLEAR ERROR SEEN FLAG
011224  010205                  MOV     R2,R5           ;GET EXPD BUFFER ADDRESS
011226  011264  002320'  15$:    MOV     (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
011232  011164  002464'         MOV     (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
011236  022221                  CMP     (R2)+,(R1)+     ;EXPD EQUAL RECV?
011240  001401                  BEQ     25$             ;BR IF YES
011242  005203                  INC     R3              ;SET ERROR SEEN FLAG
011244  062704  000002  25$:    ADD     #2,R4           ;POINT TO NEXT WORD ADDRESS
011250  020427  000014          CMP     R4,#14          ;DONE FIRST 7 WORDS?
011254  003764                  BLE     15$             ;BR IF NO
011256  032765  000200  000012  BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
011264  001403                  BEQ     50$             ;BR IF NO
011266  020427  000016          CMP     R4,#16          ;DONE EXTENDED FEATURES WORD?
011272  003755                  BLE     15$             ;BR IF NO
011274  005703          50$:    TST     R3              ;ANY ERRORS SEEN?
011276  001402                  BEQ     55$             ;BR IF NO
011300  000241                  CLC                     ;SET FAILURE
011302  000401                  BR      60$             ;
011304  000261          55$:    SEC                     ;SET SUCCESS
011306  000207          60$:    RTS     PC              ;RETURN


                        .SBTTL  CKMSG2  - COMPARE EXPD RECV MESSAGE BUFFERS
```

```
                                ;+
                                ;
                                ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                                ;ERROR PRINT ROUTINES.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
                                ;       R3      NUMBER OF BYTES TO COMPARE
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
                                ;               CLR - MESSAGE BUFFERS DON'T MATCH
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
                                ;       RECMSG          BUFFER IS SET TO RECV DATA
                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                                ;
                                ;-
011310                          CKMSG2::
                                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011310  020327  000144                  CMP     R3,#RECMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
011314  003412                          BLE     5$                      ;@@D BR IF NO
011320  012703  000144                  MOV     #RECMSG-EXPMSG,R3;@@D
011322                                  PRINTF  #DEBUGMSG               ;@@D
011326  012746  011442'                  MOV     #DEBUGMSG,-(SP)
011332  012746  000001                  MOV     #1,-(SP)
011336  010600                          MOV     SP,R0
011340  104417                          TRAP    C$PNTF
011342  062706  000004                  ADD     #4,SP
011346  010037  002302'          5$:     MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
011352  010137  002304'                  MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
011356  005737  003132'                  TST     KTENABLE                ;TESTING ABOVE 28K?
011362  001403                          BEQ     10$                     ;BR IF NO
011364  004737  017130'                  JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
011370  010001                          MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
011372  005004          10$:    CLR     R4                      ;WORD IN BUFFER
011374  005005                  CLR     R5                      ;CLEAR ERROR SEEN FLAG
011376  111264  002320'    15$:    MOVB    (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
011402  111164  002464'            MOVB    (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
011406  122221                  CMPB    (R2)+,(R1)+             ;EXPD EQUAL RECV?
011410  001401                  BEQ     25$                     ;BR IF YES
011412  005205                  INC     R5                      ;SET ERROR SEEN FLAG
011414  062704  000001    25$:    ADD     #1,R4                   ;POINT TO NEXT BYTE
011420  020403                  CMP     R4,R3                   ;DONE ALL BYTES?
011422  002001                  BGE     50$                     ;BR IF YES
011424  000764                  BR      15$                     ;DO NEXT BYTE
011426  005705          50$:    TST     R5                      ;ANY ERRORS SEEN?
011430  001402                  BEQ     55$                     ;BR IF NO
011432  000241                  CLC                             ;SET FAILURE
```

```
1434 011434  000401                        BR      60$             ;
1435 011436  000261                55$:    SEC                     ;SET SUCCESS
1436 011440  000207                60$:    RTS     PC              ;RETURN
1437
1438 011442      120      122      117  DEBUGMSG:       .ASCIZ  'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';@@D
1439 011532      045      116      045  FERCM:  .ASCII  /%N%A  ***/
1440 011543      040      040      124  ERCM:   .ASCIZ  /  TSSR ERROR CODE REC'D = /
1441 011576      056      056      056  SIMSG:  .ASCIZ  /.... AFTER DOING SOFT INIT/
1442 011631      124      105      123  TINERR: .ASCIZ  /TEST: .../
1443                                            .EVEN
1444
1445
1446                                    ;*
1447                                    ;
1448                                    ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1449                                    ;
1450                                    ;INPUT:
1451                                    ;
1452                                    ;       R1      CONTENTS OF TSSR AT ERROR
1453                                    ;
1454                                    ;SIDE EFFECTS:
1455                                    ;
1456                                    ;       EXECUTES DROP UNIT TO CEASE TESTING
1457                                    ;
1458                                    ;-
1459
1460 011644                                     BGNMSG  SFIMSG
     011644                             SFIMSG::
1461 011644  004737  005634'                    JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1462 011650  004737  017014'                    JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
1463 011654                                     ENDMSG
     011654                             L10003:
     011654  104423                            TRAP    C$MSG
1464
1465                                    ;*
1466                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1467                                    ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
1468                                    ;
1469                                    ;INPUTS:
1470                                    ;
1471                                    ;       R1      TSSR CONTENTS
1472                                    ;       R4      ADDRESS OF COMMAND PACKET
1473                                    ;
1474                                    ;-
1475
1476 011656                                     BGNMSG  PKTSSR
     011656                             PKTSSR::
1477 011656  004737  005634'                    JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1478 011662  012700  000004                    MOV     #4,R0           ;NO. OF WORDS IN PACKET
1479 011666  004737  007200'                    JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
1480 011672                                     ENDMSG
     011672                             L10004:
     011672  104423                            TRAP    C$MSG
1481
1482                                    ;*
1483                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1484                                    ;TSSR AND A GET STATUS COMMAND PACKET.
```

```
                        ;
                        ;INPUTS:
                        ;
                        ;       R1      TSSR CONTENTS
                        ;       R4      ADDRESS OF COMMAND PACKET
                        ;
                        ;-

                                BGNMSG  PKTGETS
011674                  PKTGETS::
011674  004737  005634' 1494            JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
011700  012700  000002 1495            MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
011704  004737  007200' 1496            JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
011710                  1497            ENDMSG
011710                  L10005:
011710  104423          1498            TRAP    C$MSG
                        1499
                        1500
                        1501    ;+
                        1502    ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
                        1503    ;
                        1504    ;INPUTS:
                        1505    ;
                        1506    ;       R1      TSSR CONTENTS
                        1507    ;       R4      ADDRESS OF COMMAND PACKET
                        1508    ;-
                        1509
                                BGNMSG  SFFMSG
011712                  SFFMSG::
011712  004737  005634' 1510            JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
011716                  1511            ENDMSG
011716                  L10006:
011716  104423          1512            TRAP    C$MSG
                        1513
                        1514
                        1515            .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
                        1516    ;+
                        1517    ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
                        1518    ;BUFFER FOR ERROR REPORTS
                        1519    ;
                        1520    ;INPUTS:
                        1521    ;
                        1522    ;       R1      CONTENTS OF TSSR
                        1523    ;       R2      LOW ORDER MESSAGE BUFFER
                        1524    ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
                        1525    ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
                        1526    ;-
                        1527
                                BGNMSG  PKTMES
011720                  PKTMES::
011720  004737  005634' 1528            JSR     PC,PRITSSR              ;PRINT CONTENTS OF TSSR
011724  010200          1529            MOV     R2,R0                   ;LOW ORDER ADDRESS
011726  010301          1530            MOV     R3,R1                   ;HIGH ORDER ADDRESS
011730  004737  014052' 1531            JSR     PC,PRMESS               ;PRINT THE MESSAGE BUFFER
011734                  1532            ENDMSG
011734                  L10007:
011734  104423          1533            TRAP    C$MSG
```

```
                                        .SBTTL  ADDSSR  - PRINT TEST ADDRESS AND TSSR
                                ;+
                                ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                ;TSSR AND A MEMORY TEST ADDRESS
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       ERRHI   HIGH ORDER MEMORY TEST ADDRESS
                                ;       ERRLO   LOW ORDER MEMORY TEST ADDRESS
                                ;-
                                        BGNMSG  ADDSSR
011736                          ADDSSR::
011736
011736  004737  010104'                JSR     PC,PRTTOD       ;PRINT MEMORY TEST ADDRESS
011742  016501  000002                  MOV     TSSR(R5),P1     ;GET CURRENT TSSR
011746  004737  005634'                JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
011752                                  ENDMSG
011752                          L10010:
011752  104423                          TRAP    C$MSG


                                        .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
                                        BGNMSG  MSGEXP
011754                          MSGEXP::
011754
011754  012700  000007                  MOV     #7,R0           ;ASSUME NO EXT FEATURES
011760  005737  002224'                TST     EXTFEA          ;EXT FEATURES SET?
011764  001402                          BEQ     5$              ;BR IF NO
011766  012700  000010                  MOV     #8.,R0          ;EXT FEATURE BUFFER IS 8 WORDS
011772  004737  014362'        5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
011776                                  ENDMSG
011776                          L10011:
011776  104423                          TRAP    C$MSG


                                        .SBTTL  FIFEXP  - PRINT FIFO EXP/RECV DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
                                ;
                                ;       R1      - BYTE COUNT
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY
```

```
1619 012222  012240' 012302' 012373' STATCOD:        .WORD   1$,2$,3$,4$,5$,6$,0
1620 012240     045     116     045  1$:.ASCIZ  '<N><A Tape Bus Signals in Word #8:'
1621 012302     045     116     045  2$:.ASCIZ  '<N><A         PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>'
1622 012373     045     116     045  3$:.ASCIZ  '<N><A         IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>'
1623 012464     045     116     045  4$:.ASCIZ  '<N><A         IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>'
1624 012555     045     116     045  5$:.ASCIZ  '<N><A Tape Bus Signals in Word #9:'
1625 012617     045     116     045  6$:.ASCIZ  '<N><A         DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>'
1626                                          .EVEN
1627
1628
1629
1630                                          .SBTTL  MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
1631                                  ;+
1632                                  ;
1633                                  ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
1634                                  ;
1635                                  ;IMPLICIT INPUTS:
1636                                  ;
1637                                  ;       EXPMSG  - EXPECTED MESSAGE BUFFER
1638                                  ;       RECMSG  - RECEIVED MESSAGE BUFFER
1639                                  ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1640                                  ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1641                                  ;-
1642 012674                                      BGNMSG  MSGLOOP
     012674                          MSGLOOP::
1643 012674  012701  012736'                     MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
1644 012700  012100                  10$:        MOV     (R1)+,R0        ;DONE ALL MSG LINES?
1645 012702  001410                              BEQ     20$             ;BR IF YES
1646 012704                                      PRINTX  R0              ;PRINT STATUS BIT NAMES
     012704  010046                              MOV     R0,-(SP)
     012706  012746  000001                      MOV     #1,-(SP)
     012712  010600                              MOV     SP,R0
     012714  104415                              TRAP    C$PNTX
     012716  062706  000004                      ADD     #4,SP
1647 012722  000766                              BR      10$             ;DO ANOTHER MSG LINE
1648 012724  012700  000012          20$:        MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
1649 012730  004737  014362'                     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
1650 012734                                      ENDMSG
     012734                          L10014:
     012734  104423                              TRAP    C$MSG
1651
1652 012736  012756' 013031' 013130' LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
1653 012756     045     116     045  1$:.ASCIZ  '<N><A Tape Bus Loopback Signals in Word #8:'
1654 013031     045     116     045  2$:.ASCIZ  '<N><A       PARERR<15>              IRESV2<14>              IRESV1<13>'
1655 013130     045     116     045  3$:.ASCIZ  '<N><A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
1656 013227     045     116     045  4$:.ASCIZ  '<N><A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
1657 013326     045     116     045  5$:.ASCIZ  '<N><A ITAD0=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
1658 013425     045     116     045  6$:.ASCIZ  '<N><A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
1659 013524     045     116     045  7$:.ASCIZ  '<N><A IGO  =>IFPT<00>'
1660                                          .EVEN
1661
1662                                          .SBTTL  MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
1663                                  ;+
1664                                  ;
1665                                  ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
1666                                  ;
1667                                  ;
```

```
                                        ;IMPLICIT INPUTS:
                                        ;
                                        ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                        ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                        ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                        ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                        ;-
                                                BGNMSG  MSGSUB
013552                                  MSGSUB::
013552                                          MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
013552  012700  000012                    
013556  004737  014362'                  
013562                                          JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
                                                ENDMSG
013562                                  L10015:
013562  104423                                  TRAP    C$MSG
```

```
                                                .SBTTL  MEMADD - PRINT MEMORY ADDRESS DATA ERROR
                                        ;+
                                        ;
                                        ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
                                        ;
                                        ;IMPLICIT INPUTS:
                                        ;
                                        ;       ERRHI   - MEMORY ERROR HIGH ORDER ADDRESS
                                        ;       ERRLO   - MEMORY ERROR LOW ORDER ADDRESS
                                        ;       EXP     - EXPECTED DATA
                                        ;       RECV    - RECEIVED DATA
                                        ;-
                                                BGNMSG  MEMADD
013564                                  MEMADD::
013564                                          JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
013564  004737  007770'                  
013570  013701  002230'                          MOV     EXPD,R1         ;GET EXPD DATA
013574  013702  002232'                          MOV     RECV,R2         ;GET RECEIVED DATA
013600  004737  007552'                          JSR     PC,PRIXOR       ;PRINT EXPD/RECV
013604                                          ENDMSG
013604                                  L10016:
013604  104423                                  TRAP    C$MSG
```

```
                                                .SBTTL  PRAMPKT - PRINT RAM AND PACKET DATA
                                        ;+
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                        ;WHEN THE RAM DATA DOES NOT MATCH.
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R4      POINTER TO COMMAND PACKET
                                        ;
                                        ;IMPLICIT INPUTS:
                                        ;
                                        ;       RAMDATA         DATA AS READ FROM THE RAM
                                        ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                        ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                        ;
```

```
                                        ;IMPLICIT OUTPUTS:
                                        ;
                                        ;       RAMSIZ  SET TO 0
                                        ;-

                                        PRAMPKT:
013606                                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
013606 012701  002240'                          MOV     #RAMDATA,R1             ;DATA FROM THE RAM
013612 005002                                   CLR     R2                      ;INIT BYTE NUMBER
013616 122124                           5$:     CMPB    (R1)+,(R4)+             ;COMPARE EXPECTED, RECEIVED
013620 001005                                   BNE     7$                      ;BR IF NO MATCH
013622                                          FORCERROR       7$,NOTSSR
013624 000436                                   BR      10$                     ;BRD
013626 116105  177777                   7$:     MOVB    -1(R1),R5               ;GET RECV RAM DATA
013634 116403  177777                           MOVB    -1(R4),R3               ;GET EXPD PACKET DATA
013642                                          XOR     R5,R3                   ;XOR EXPD/RECV
013646 042703  177400                           BIC     #177400,R3              ;LOW BYTE ONLY
013656 116137  177777  002232'                   MOVB    -1(R1),RECV             ;GET RECEIVED RAM DATA
013662 116437  177777  002230'                   MOVB    -1(R4),EXPD             ;GET EXPECTED RAM DATA
013670                                          PRINTB  #RAMASC,R2,RECV,EXPD,R3
013676 010346                                   MOV     R3,-(SP)
013700 013746  002230'                          MOV     EXPD,-(SP)
013704 013746  002232'                          MOV     RECV,-(SP)
013710 010246                                   MOV     R2,-(SP)
013712 012746  013766'                          MOV     #RAMASC,-(SP)
013716 012746  000005                           MOV     #5,-(SP)
013722 010600                                   MOV     SP,R0
013724 104414                                   TRAP    C$PNTB
013726 062706  000014                           ADD     #14,SP
013732 005202                           10$:    INC     R2                      ;UPDATE BYTE COUNT
013736 005737  002300'                          TST     RAMSIZ                  ;DEFAULT TO 8.?
013742 001404                                   BEQ     15$                     ;BR IF YES
013744 020237  002300'                          CMP     R2,RAMSIZ               ;DONE ALL BYTES?
013750 003724                                   BLE     5$                      ;BR IF NO
013752 000403                                   BR      25$                     ;
013754 020227  000010                   15$:    CMP     R2,#8.                  ;DONE DEFAULT NUMBER OF BYTES?
013760 002720                           20$:    BLT     5$                      ;BR IF NO
013762 005037  002300'                  25$:    CLR     RAMSIZ                  ;SET DEFAULT RAMSIZ
013766 000207                                   RTS     PC                      ;RETURN

013766  045  116  045  RAMASC: .ASCIZ  '%N%A  BYTE: %D2%A RAM: %03%A Packet: %03%A XOR: %03'
                              .EVEN

                                        .SBTTL  PRMESS  - PRINT CONTENTS OF MESSAGE BUFFER
                                ;+
                                ;
                                ;THIS ROUTINE PRINTS THE CONTENTS OF
                                ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
                                ;TSV-05.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
                                ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
                                ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                ;
                                ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
```

```
                                        ;
                                        ;-
                                        ;
014052                                  PRMESS:
014052                                          SAVREG                  ;SAVE THE REGISTERS
014056  010005                                  MOV     R0,R5           ;SAVE LOW ORDER ADDRESS
014060  005737  003132'                         TST     KTENABLE        ;ADDRESS ABOVE 28K?
014064  001001                                  BNE     10$             ;BR IF YES
014066  005001                                  CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
014070  010103                          10$:    MOV     R1,R3           ;SAVE HIGH ORDER ADDRESS
014072  006100                                  ROL     R0              ;SHIFT BIT15 TO C BIT
014074  006101                                  ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
014076                                          PRINTX  #PROASC,R1,R5   ;PRINT MESSAGE BUFFER ADDRESS
014076  010546                                  MOV     R5,-(SP)
014100  010146                                  MOV     R1,-(SP)
014102  012746  014230'                         MOV     #PROASC,-(SP)
014106  012746  000003                         MOV     #3,-(SP)
014112  010600                                  MOV     SP,R0
014114  104415                                  TRAP    C$PNTX
014116  062706  000010                         ADD     #10,SP
014122                                          PRINTX  #PR1ASC         ;PRINT HEADER FOR CONTENTS
014122  012746  014275'                         MOV     #PR1ASC,-(SP)
014126  012746  000001                         MOV     #1,-(SP)
014132  010600                                  MOV     SP,R0
014134  104415                                  TRAP    C$PNTX
014136  062706  000004                         ADD     #4,SP
014142  005004                                  CLR     R4              ;NUMBER OF THE NEXT WORD
014144  010501                                  MOV     R5,R1           ;COPY LOW ORDER ADDRESS
014146  010300                                  MOV     R3,R0           ;COPY HIGH ORDER ADDRESS
014150  001405                                  BEQ     20$             ;BR IF NOT ABOVE 28K
014152  004737  017130'                         JSR     PC,SETMAP       ;SETUP PAR ADDRESS IN R0
014156  010005                                  MOV     R0,R5           ;GET PAR FORMAT ADDRESS ABOVE 28K
014160                                  20$:    PRINTX  #PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
014160  012546                                  MOV     (R5)+,-(SP)
014162  010446                                  MOV     R4,-(SP)
014164  012746  014333'                         MOV     #PRASC,-(SP)
014170  012746  000003                         MOV     #3,-(SP)
014174  010600                                  MOV     SP,R0
014176  104415                                  TRAP    C$PNTX
014200  062706  000010                         ADD     #10,SP
014204  005204                                  INC     R4              ;NUMBER OF THE NEXT
014206  020427  000007                         CMP     R4,#7           ;DONE ALL YET ?
014212  003005                                  BGT     50$             ;BRANCH IF ALL DONE
014214  002761                                  BLT     20$             ;PRINT FIRST 7 WORDS
014216  032763  000200  000012                 BIT     #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
SET
014224  001355                                  BNE     20$             ;PRINT EXTENDED STATUS WORD
014226  000207                          50$:    RTS     PC              ;RETURN
                                        ;
014230      045     116     045         PROASC: .ASCIZ  '%N%A Message Buffer Address - %O1%05'
014275      045     116     045         PR1ASC: .ASCIZ  '%N%A Message Buffer Contents:'
014333      045     116     045         PRASC:  .ASCIZ  '%N%A    Word%D1%A: %O'
                                                .EVEN
                                        ;
                                                .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                                        ;+
                                        ;
                                        ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
```

```
                                        ;
                                        ;       R0      - NUMBER OF WORDS IN BUFFER
                                        ;
                                        ;IMPLICIT INPUTS:
                                        ;
                                        ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                        ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                        ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                        ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                        ;-
014362                                  PRMSGEXP::
014362                                          SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
014366  010005                                  MOV     R0,R5           ;SAVE NUMBER OF WORDS
014370  013700  002304'                          MOV     RCVLOADD,R0     ;GET RECV LOW ADDRESS
014374  010004                                  MOV     R0,R4           ;COPY LOW ADDRESS
014376  013701  002302'                          MOV     RCVHIADD,R1     ;GET RECV HIGH ADDRESS
014402  006100                                  ROL     R0              ;SHIFT BIT15 TO C BIT
014404  006101                                  ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
014406                                          PRINTX  #PRMSG0,R1,R4   ;PRINT MESSAGE BUFFER ADDRESS
014406  010446                                  MOV     R4,-(SP)
014410  010146                                  MOV     R1,-(SP)
014412  012746  014542'                          MOV     #PRMSG0,-(SP)
014416  012746  000003                          MOV     #3,-(SP)
014422  010600                                  MOV     SP,R0
014424  104415                                  TRAP    C$PNTX
014426  062706  000010                          ADD     #10,SP
014432                                          PRINTX  #PRMSG1         ;PRINT HEADER FOR CONTENTS
014432  012746  014607'                          MOV     #PRMSG1,-(SP)
014436  012746  000001                          MOV     #1,-(SP)
014442  010600                                  MOV     SP,R0
014444  104415                                  TRAP    C$PNTX
014446  062706  000004                          ADD     #4,SP
014452  005004                                  CLR     R4              ;NUMBER OF THE CURRENT WORD
014454  012701  002320'                          MOV     #EXPMSG,R1      ;GET EXPD BUFFER ADDRESS
014460  012702  002464'                          MOV     #RECMSG,R2      ;GET RECV BUFFER ADDRESS
014464  011100                        20$:      MOV     (R1),R0         ;GET EXPD
014466  011203                                  MOV     (R2),R3         ;GET RECV
014470                                          XOR     R0,R3           ;XOR EXPD/RECV
014500                                          PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
014500  010346                                  MOV     R3,-(SP)
014502  012246                                  MOV     (R2)+,-(SP)
014504  012146                                  MOV     (R1)+,-(SP)
014506  010446                                  MOV     R4,-(SP)
014510  012746  014645'                          MOV     #PRMSG2,-(SP)
014514  012746  000005                          MOV     #5,-(SP)
014520  010600                                  MOV     SP,R0
014522  104415                                  TRAP    C$PNTX
014524  062706  000014                          ADD     #14,SP
014530  005204                                  INC     R4              ;NUMBER OF THE NEXT
014532  020405                                  CMP     R4,R5           ;DONE ALL YET?
014534  002001                                  BGE     50$             ;BR IF YES
014536  000752                                  BR      20$             ;DO ANOTHER
014540  000207                        50$:      RTS     PC              ;RETURN
                                        ;
014542      045     116     045       PRMSG0: .ASCIZ  '%N%A Message Buffer Address - %01%05
014607      045     116     045       PRMSG1: .ASCIZ  '%N%A Message Buffer Contents:'
014645      045     116     045       PRMSG2: .ASCIZ  '%N%A   WORD %%D2%A  EXPD: %06%A  RECV: %06%A  XOR: %06'
```

```
                        .EVEN

                        .SBTTL  PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
                ;+
                ;
                ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
                ;       ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
                ;
                ;       R0      - NUMBER OF BYTES IN BUFFER
                ;
                ;IMPLICIT INPUTS:
                ;
                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                ;-
014732          PRBYTEXP::
014732                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014736  010005          MOV     R0,R5                   ;SAVE NUMBER OF BYTES
014740  005037  002316' CLR     PRMNO                   ;INIT ERROR COUNT
014744  005004          CLR     R4                      ;NUMBER OF THE CURRENT BYTE
014746  012701  002320' MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
014752  012702  002464' MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
014756  111100          20$:    MOVB    (R1),R0         ;GET EXPD BYTE
014760  042700  177400          BIC     #^C<377>,R0     ;CLEAR UPPER BYTE
014764  110037  015300' MOVB    R0,PRBEXP               ;SAVE FOR ERROR REPORT
014770  111203          MOVB    (R2),R3                 ;GET RECV BYTE
014772  042703  177400          BIC     #^C<377>,R3     ;CLEAR UPPER BYTE
014776  110337  015302' MOVB    R3,PRBREC               ;FOR ERROR REPORT
015002          XOR     R0,R3                           ;XOR EXPD/RECV
015012  122122          CMPB    (R1)+,(R2)+             ;EXPD = RECV?
015014  001431          BEQ     30$                     ;BR IF YES
015016  005237  002316' INC     PRMNO                   ;UPDATE ERROR COUNT
015022  023727  002316' 000010 CMP     PRMNO,#8.       ;PRINTED 8?
015030  101023          BHI     30$                     ;BR IF YES
015032                  27$:    PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
015032  010346          MOV     R3,-(SP)
015034  013746  015302' MOV     PRBREC,-(SP)
015040  013746  015300' MOV     PRBEXP,-(SP)
015044  010446          MOV     R4,-(SP)
015046  012746  015146' MOV     #PRBMSG,-(SP)
015052  012746  000005          MOV     #5,-(SP)
015056  010600          MOV     SP,R0
015060  104415          TRAP    C$PNTX
015062  062706  000014          ADD     #14,SP
015066                  FORCEXIT        50$             ;BUG
015076  000404          BR      35$                     ;BUG
015100                  30$:
015100                  FORCERROR  27$,NOTSSR           ;BUG
015110                  35$:                            ;BUG
015110  005204          INC     R4                      ;NUMBER OF THE NEXT
015112  020405          CMP     R4,R5                   ;DONE ALL YET?
015114  002001          BGE     50$                     ;BR IF YES
015116  000717          BR      20$                     ;DO ANOTHER
015120                  50$:    PRINTX  #PRBTOT,PRMNO   ;PRINT TOTAL ERROR COUNT
015120  013746  002316' MOV     PRMNO,-(SP)
015124  012746  015233' MOV     #PRBTOT,-(SP)
015130  012746  000002          MOV     #2,-(SP)
```

```
                015134  010600                          MOV     SP,R0
                015136  104415                          TRAP    C$PNTX
                015140  062706  000006                  ADD     #6,SP
          1886  015144  000207                          RTS     PC              ;RETURN
          1887
          1888  015146     045     116     045  PRBMSG: .ASCIZ  '#N#A   BYTE #D2#A  EXPD: #03#A  RECV: #03#A  XOR: #03'
          1889  015233     045     116     045  PRBTOT: .ASCIZ  '#N#A NUMBER OF BYTES IN ERROR = #D2'
          1890                                          .EVEN
          1891  015300  000000                  PRBEXP: .WORD   0               ;EXPD
          1892  015302  000000                  PRBREC: .WORD   0               ;RECV
          1893
          1894                                          .SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
          1895                                  ;+
          1896                                  ;
          1897                                  ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
          1898                                  ;
          1899                                  ;INPUTS:
          1900                                  ;
          1901                                  ;       R1      RECEIVED DATA
          1902                                  ;       R2      EXPECTED DATA
          1903                                  ;
          1904                                  ;-
          1905
          1906  015304                                  BGNMSG  EXPREC
                015304                          EXPREC::
          1907  015304  004737  007552'                JSR     PC,PRIXOR               ;PRINT THE DATA
          1908  015310                                  ENDMSG
                015310                          L10017:
                015310  104423                          TRAP    C$MSG
          1909
          1910
          1911
          1912
          1913                                          .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
          1914                                  ;+
          1915                                  ;
          1916                                  ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
          1917                                  ;
          1918                                  ;
          1919                                  ;INPUTS:
          1920                                  ;
          1921                                  ;       R1      RECEIVED DATA BYTE
          1922                                  ;       R2      EXPECTED DATA BYTE
          1923                                  ;
          1924                                  ;-
          1925
          1926  015312                                  BGNMSG  EXPBREC
                015312                          EXPBREC::
          1927  015312  004737  007422'                JSR     PC,PRIBXOR              ;PRINT THE DATA
          1928  015316                                  ENDMSG
                015316                          L10020:
                015316  104423                          TRAP    C$MSG
          1929
          1930
          1931
          1932
          1933                                          .SBTTL  RAMERR  - PRINT RAM AND PACKET DATA
```

```
                        ;+
                        ;
                        ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                        ;
                        ;INPUTS:
                        ;
                        ;       R4      POINTER TO COMMAND PACKET
                        ;
                        ;IMPLICIT INPUTS:
                        ;
                        ;       RAMDATA         DATA AS READ FROM THE RAM
                        ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                        ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                        ;
                        ;IMPLICIT OUTPUTS:
                        ;
                        ;       RAMSIZ  SET TO 0
                        ;-

                        BGNMSG  RAMERR
                RAMERR::
015320  004737  013606' JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                        ENDMSG
                L10021:
015324  104423         TRAP    C$MSC


                        .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
                        ;+
                        ;
                        ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                        ;
                        ;INPUTS:
                        ;
                        ;       R4      POINTER TO COMMAND PACKET
                        ;
                        ;IMPLICIT INPUTS:
                        ;
                        ;       RAMDATA         DATA AS READ FROM THE RAM
                        ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                        ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                        ;       ERRHI           HIGH ORDER TEST ADDRESS
                        ;       ERRLO           LOW ORDER TEST ADDRESS
                        ;
                        ;IMPLICIT OUTPUTS:
                        ;
                        ;       RAMSIZ  SET TO 0
                        ;-

                        BGNMSG  RAMTADD
                RAMTADD::
015326  004737  010104' JSR     PC,PRITADD      ;PRINT TEST ADDRESS
015332  004737  013606' JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                        ENDMSG
                L10022:
015336  104423         TRAP    C$MSG
```

```
                                        .SBTTL  RAMEXP  - PRINT RAM EXPD/RECV DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;       R4      CONTROLLER RAM ADDRESS
                                ;-
                                        BGNMSG  RAMEXP
                                RAMEXP::
                                        BIC     #^C<377>,R1             ;SAVE EXPD RAM DATA BYTE
                                        BIC     #^C<377>,R2             ;SAVE EXPD RAM DATA BYTE
                                        JSR     PC,PRIRAM               ;PRINT THE RAM ADDRESS
                                        JSR     PC,PRIXOR               ;PRINT THE DATA
                                        ENDMSG
                                L10023:
                                        TRAP    C$MSG

                                        .SBTTL  TIMEXP  - PRINT TIMER A,B AND EXP/REC
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                ;AND TIMER A,B HEADER MESSAGE
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;-
                                        BGNMSG  TIMEXP
                                TIMEXP::
                                        PRINTX  #TIMSGO                 ;PRINT HEADER
                                        MOV     #TIMSGO,-(SP)
                                        MOV     #1,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTX
                                        ADD     #4,SP
                                        JSR     PC,PRIXOR               ;PRINT THE DATA
                                        ENDMSG
                                L10024:
                                        TRAP    C$MSG

015410  045  116  045  TIMSGO: .ASCIZ  '<N><A TIMER A STATUS IS IN BIT 3<N><A TIMER B STATUS IS IN BIT 2'
                                .EVEN


                                        .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

                                ;+
                                ;
```

```
                                     ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
                                     ;
                                     ;INPUTS:
                                     ;
                                     ;       R1      CONTENTS OF TSSR
                                     ;       R2      DATA WRITTEN (8 BITS)
                                     ;
                                     ;-

015510                                       BGNMSG  BADSSR
015510                               BADSSR::
015510  010246                               MOV     R2,-(SP)                ;SAVE DATA TRANSFERRED
015512  042702  177400                       BIC     #177400,R2              ;GET JUST ONE BYTE
015516                                       PRINTB  #XFERASC,R2
015516  010246                               MOV     R2,-(SP)
015520  012746  015550'                      MOV     #XFERASC,-(SP)
015524  012746  000002                       MOV     #2,-(SP)
015530  010600                               MOV     SP,R0
015532  104414                               TRAP    C$PNTB
015534  062706  000006                       ADD     #6,SP
015540  012602                               MOV     (SP)+,R2                ;RESTORE R2
015542  004737  005634'                      JSR     PC,PRITSSR              ;DECODE TSSR CONTENTS
015546                                       ENDMSG
015546                               L10025:
015546  104423                               TRAP    C$MSG
015550     045     116     045       XFERASC:        .ASCIZ  '%N%A Data Transferred = %O3'


                                             .SBTTL  GLOBAL SUBROUTINES SECTION

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

                                             .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER

                                     ;+
                                     ;
                                     ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
                                     ;BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
                                     ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
                                     ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
                                     ;
                                     ;INPUTS:
                                     ;
                                     ;       R5      ADDRESS OF FIRST REGISTER
                                     ;
                                     ;OUTPUTS:
                                     ;
                                     ;       R0      CONTENTS OF TSSR, IF ERROR
                                     ;       CARRY   SET IF INIT WAS OKAY
                                     ;               CLEAR IF FATAL ERROR
                                     ;
                                     ;CALLING SEQUENCE:
                                     ;
                                     ;       MOV     #ADDRESS,R5
```

```
2079                                    ;       JSR     PC,SOFINIT
2080                                    ;       BCS     CONTINUE
2081                                    ;       ERRDF                   ;REPORT FATAL ERROR
2082                                    ;
2083                                    ;-
2084
2085 015604                            SOFINIT::
2086 015604                                    SAVREG                  ; SAVE THE REGISTERS
2087 015610  012765  000000  000002             MOV     #0,TSSR(R5)     ; DO THE INIT.
2088 015616  004737  016060'                    JSR     PC,WAITF        ; WAIT FOR SSR
2089 015622  016500  000002                     MOV     TSSR(R5),R0     ;GET THE TSSR REGISTER
2090 015626  010004                             MOV     R0,R4           ;TSSR CONTENTS
2091 015630  042704  176277                     BIC     #^C<HIADDR!OFL>,R4
2092 015634  052704  002200                     BIS     #SSR!NBA,R4     ;R4 HAS EXPECTED CONTENTS
2093 015640  020400                             CMP     R4,R0           ;ONLY EXPECTED BITS SET ?
2094 015642  001402                             BEQ     5$              ;BRANCH IF OKAY
2095 015644  000241                             CLC                     ;CLEAR THE CARRY FOR ERROR
2096 015646  000401                             BR      10$             ;GO TO EXIT
2097 015650  000261                    5$:      SEC                     ;SET THE CARRY BIT
2098 015652  000207                    10$:     RTS     PC              ;RETURN TO CALLER
2099
2100                                            .SBTTL  CHKAMB  - CHECK TSSR FOR AMBIGUITY
2101
2102                                    ;+
2103                                    ;
2104                                    ;THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
2105                                    ;FOR AMBIGUITY
2106                                    ;
2107                                    ;INPUT:
2108                                    ;
2109                                    ;       R0      CONTENTS OF TSSR
2110                                    ;
2111                                    ;OUTPUT:
2112                                    ;
2113                                    ;       R0      CONTENTS OF TSSR
2114                                    ;
2115                                    ;       CARRY   SET - NO AMBIGUITY
2116                                    ;               CLR - AMBIGUOUS CONTENTS
2117                                    ;
2118                                    ;-
2119
2120 015654                            CHKAMB:
2121 015654                                    SAVREG                  ;SAVE THE GENERAL REGISTERS
2122 015660  010004                             MOV     R0,R4           ;CONTENTS OF TSSR
2123 015662  032700  100000                     BIT     #SC,R0          ;IS BIT 15 SET ?
2124 015666  001004                             BNE     5$              ;BRANCH IF YES
2125 015670  032700  174077                     BIT     #^C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
2126 015674  001023                             BNE     40$             ;MUST BE AN ERROR
2127 015676  000424                             BR      45$             ;RETURN WITH SUCCESS
2128 015700  032700  000200            5$:      BIT     #SSR,R0         ;IS READY BIT SET ?
2129 015704  001011                             BNE     10$             ;BRANCH IF READY BIT IS SET.
2130 015706  032700  000040                     BIT     #BIT5,R0        ;IS FATAL ERROR BIT SET ?
2131 015712  001414                             BEQ     40$             ;ERROR IF NOT
2132 015714  042704  177761                     BIC     #CTERCLS,R4     ;CLEAR ALL BUT TERMINATION CODE
2133 015720  020427  000016                     CMP     R4,#16          ;ALL THREE BITS MUST BE SET
2134 015724  001007                             BNE     40$             ;ERROR IF NOT SET
2135 015726  000410                             BR      45$             ;OK IF ALL ARE SET
```

```
2136 015730  032700  000040          10$:    BIT     #BIT5,R0        ;IS FATAL ERROR BIT SET ?
2137 015734  001405                          BEQ     45$             ;ERROR IF BIT IS SET WITH SSR
2138 015736  032700  000006                  BIT     #BIT2!BIT1,R0   ;IS THIS A FUNCTION REJECT
2139 015742  001002                          BNE     45$             ;BR, IF TSSR IS OK
2140 015744  000241                  40$:    CLC                     ;AMBIGUOUS CONTENTS
2141 015746  000401                          BR      50$
2142 015750  000261                  45$:    SEC                     ;SHOW SUCCESS - NO AMBIGUITY
2143 015752  000207                  50$:    RTS     PC              ;RETURN TO CALLER
2144
2145                                          .SBTTL  ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
2146                                  ;
2147                                  ; DEFAULT DISPLAY INTERRUPT HANDLERS.
2148                                  ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2149                                  ; OTHERWISE, SAVE CPU REGISTERS AND DISMISS.
2150                                  ;
2151
2152                                  ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2153                                  ;
2154          000200                          IOKCKIN=BIT7    ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
2155          000001                          IOKSTP=BIT0     ; EXPECT "STOP" INTERRUPT.
2156                                  ;
2157                                  ;INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2158 015754      000                 INTMASK:        .BYTE   0
2159                                  ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2160 015755      000                 INTFLAG:        .BYTE   0
2161
2162                                  ;SAVED INTERRUPT VECTOR:
2163 015756  000000                  INTVEC: .WORD   0
2164                                  ;SAVE CPU PC
2165 015760  000000                  INTCPC: .WORD   0
2166
2167                                  ;SUBROUTINE TO ENABLE INTERRUPTS:
2168 015762  010046                  ENAINT: MOV     R0,-(SP)        ;SAVE R0
2169 015764  013700  002206'                  MOV     IVEC,R0         ;GET POINTER TO VECTORS
2170 015770  012720  016026'                  MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
2171 015774  012720  000340                  MOV     #PRI07,(R0)+
2172 016000  012600                          MOV     (SP)+,R0        ;RESTORE R0
2173 016002  011646                          MOV     (SP),-(SP)
2174 016004  012766  000000  000002          MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
2175 016012  000002                          RTI
2176
2177                                  ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2178 016014  011646                  DSBINT: MOV     (SP),-(SP)
2179 016016  012766  000340  000002          MOV     #PRI07,2(SP)
2180 016024  000002                          RTI
2181
2182                                          .SBTTL  INTR    - INTERRUPT HANDLERS
2183
2184 016026                                   BGNSRV  INTR            ;DEFINE INTERRUPT ENTRY
     016026                          INTR::
2185 016026  012737  000001  002222'          MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
2186 016034  105037  015755'                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2187 016040  132737  000001  015754'          BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2188 016046  001003                          BNE     1$              ;BR IF YES
2189 016050  152737  000001  015755'          BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2190
2191                                  ;SAVE REGISTERS, MSG BUFFER, ETC.
```

```
                        ;       R0      CONTENTS OF TSSR
                        ;       CARRY   SET - OKAY
                        ;               CLR - NOT READY AMBIGUOUS, OR SC SET
                        ;
                        ;-

                        CHKTSSR:
016146  004737  016060' JSR     PC,WAITF                ;WAIT FOR READY
016152  103014         BCC     20$                     ;BRANCH IF TIME OUT
016154  004737  015654' JSR     PC,CHKAMB               ;TSSR AMBIGOUS?
016160  103006         BCC     10$                     ;BR IF YES
016162  032700  100000 BIT     #SC,R0                  ;SPECIAL CONDITION SET?
016166  001405         BEQ     15$                     ;BR IF NO
016170  032700  074000 BIT     #<SCE!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
016174  001402         BEQ     15$                     ;BR IF NO
016176  000241         10$:    CLC                     ;SET FAILURE
016200  000401         BR      20$                     ;
016202  000261         15$:    SEC                     ;SET SUCCESS
016204  000207         20$:    RTS     PC              ;RETURN TO CALLER
                        .SBTTL  XNXM    - CHECK FOR NONEXISTENT MEMORY
                        ;+
                        ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
                        ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
                        ;            "C" = 0, ALL ADDRESSES OK.
                        ;
                        ;CALL:  MOV ADR1,R1
                        ;       MOV ADR2,R2
                        ;       JSR PC,NXM
                        ;       RETURN          ;TEST "C" AND PROCEED.
                        ;
016206  012737  016242' 000004  XNXM:   MOV     #2$,@#4         ; SET BUSERR VECTOR.
016214  012737  000200  000006          MOV     #PRI04,@#6
016222  005003                  CLR     R3                      ;FLAG.
016224  000241                  CLC                             ;CLEAR THE CARRY FOR NO NXM FOUND
016226  005711                  1$:     TST     (R1)            ;TEST THE ADDRESS(ES).
                                                                ;IF ANY TRAP, CONTINUE AT 2$.
016230  020102                  CMP     R1,R2                   ;OTHERWISE, CONTINUE HERE.
016232  001407                  BEQ     3$                      ;BR IF FINISHED (NO NEXM'S).
016234  062701  000002          ADD     #2,R1                   ;SET NEXT ADDRESS...
016240  000772                  BR      1$                      ;...AND CONTINUE.

016242  005103                  2$:     COM     R3              ;GOT ONE, SET FLAG...
016244  012716  016252'                MOV     #3$,(SP)
016250  000002                  RTI                             ;...AND DISMISS INTERRUPT...
016252                          3$:     CLRVEC  #4              ;...AND GIVE BACK THE VECTOR.
016252  012700  000004                  MOV     #4,R0
016256  104436                  TRAP    C$CVEC
016260  005703                  TST     R3                      ;DID WE CATCH ONE ??
016262  001401                  BEQ     .+4                     ;NO, "C" = 0, SKIP NEXT.
016264  000261                  SEC                             ;YES, "C" = 1, (R1) = NEXM ADDR.
016266  000207                  RTS     PC


                        .SBTTL  TSTLOOP - CHECK ITERATION COUNT
```

```
2293                                    ;*
2294                                    ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
2295                                    ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
2296                                    ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
2297                                    ;
2298                                    ; CALL: LOOPTO  ARG
2299                                    ;
2300 016270                            TSTLOOP::
2301 016270  005737  002166'                    TST     NOITS           ; ITERATIONS INHIBITED?
2302 016274  001006                             BNE     1$              ; YES.
2303 016276  005737  002202'                    TST     QVP             ; NO.
2304 016302  100403                             BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
2305 016304  005337  002214'                    DEC     LOOPCNT         ; BUMP LOOP COUNTER.
2306 016310  001002                             BNE     2$
2307 016312  000241                    1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
2308 016314  000401                             BR      3$
2309 016316  000261                    2$:     SEC                     ;LOOP ENABLED.
2310 016320  000207                    3$:     RTS     PC
2311
2312                                            .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
2313                                    ;*
2314                                    ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
2315                                    ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
2316                                    ; IN THE CURRENT RUN SEQUENCE.
2317                                    ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
2318                                    ;
2319                                    ;INPUT:
2320                                    ;
2321                                    ;       R0      POINTER TO TEST ID ASCIZ STRING
2322                                    ;
2323                                    ;OUTPUT:
2324                                    ;
2325                                    ;       R5      ADDRESS OF FIRST DEVICE REGISTER
2326                                    ;
2327                                    ;IMPLICIT OUTPUTS:
2328                                    ;
2329                                    ;       TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
2330                                    ;
2331                                    ;SIDE EFFECTS:
2332                                    ;
2333                                    ;       INTERRUPT LEVEL IS RAISED TO LEVEL OF
2334                                    ;       THE DEVICE UNDER TEST
2335                                    ;
2336                                    ;-
2337
2338 016322                            TSTSETUP::
2339 016322  010046                             MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
2340 016324  005037  003152'                    CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
2341 016330  005037  016570'                    CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
2342 016334  005037  005602'                    CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
2343 016340  105037  015754'                    CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
2344 016344  013700  002200'                    MOV     UNITN,R0        ; GET THE UNIT NUMBER.
2345 016350  006300                             ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
2346 016352  005737  003112'                    TST     NODEV           ; DID STARTUP FIND THE DEVICE?
2347 016356  001430                             BEQ     4$              ; BR IF YES
2348 016360  100010                             BPL     3$              ; BR IF NOT IDLE
2349 016362  052760  160000  003174'            BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
```

```
2350 016370                                        ERRDF   1,NXR,NXRERR    ; NO DEVICE HERE -- PRINT IT
     016370  104455                                TRAP    C$ERDF
     016372  000001                                .WORD   1
     016374  003732'                               .WORD   NXR
     016376  005546'                               .WORD   NXRERR
2351 016400  000407                                BR      2$
2352 016402  052760  160001  003174' 3$:          BIS     #160001,ERTABL(R0)      ; FLAG ERROR IN THE ERROR TABLE
2353 016410                                        ERRDF   2,NOINIT        ; DEVICE NOT IDLE
     016410  104455                                TRAP    C$ERDF
     016412  000002                                .WORD   2
     016414  004327'                               .WORD   NOINIT
     016416  000000                                .WORD   0
2354 016420  012737  177777  003110' 2$:          MOV     #-1,DUFLG       ; DROP THE UNIT
2355 016426                                        DODU    UNIT
     016426  013700  002200'                       MOV     UNIT,R0
     016432  104451                                TRAP    C$DODU
2356 016434                                        DOCLN                   ; ABORT THE PASS
     016434  104444                                TRAP    C$DCLN
2357 016436  000423                                BR      5$
2358
2359 016440                              4$:       RFLAGS  R0              ; GET THE OPERATOR FLAGS.
     016440  104421                                TRAP    C$RFLA
2360 016442  032700  001000                       BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
2361 016446  001412                                BEQ     1$              ; BR IF NO
2362 016450  011600                                MOV     (SP),R0         ;GET THE ID MESSAGE
2363 016452                                        PRINTF  #TNAM,R0                ;DISPLAY THE TEST ID
     016452  010046                                MOV     R0,-(SP)
     016454  012746  016516'                       MOV     #TNAM,-(SP)
     016460  012746  000002                       MOV     #2,-(SP)
     016464  010600                                MOV     SP,R0
     016466  104417                                TRAP    C$PNTF
     016470  062706  000006                       ADD     #6,SP
2364 016474  005237  002212'            1$:       INC     TSTCNT          ; BUMP TEST COUNTER.
2365 016500                                        SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
     016500  013700  002210'                       MOV     IPRI,R0
     016504  104441                                TRAP    C$SPRI
2366 016506  005726                       5$:     TST     (SP)+           ;FIX UP THE STACK
2367 016510  013705  002204'                       MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
2368 016514  000207                                RTS     PC
2369 016516     045     123     045  TNAM:        .ASCIZ  '%S%T%A Test'
2370                                               .EVEN
2371
2372                                               .SBTTL  TSTEND  - PRINT ERRORS RECEIVED
2373                                       ;
2374                                       ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2375                                       ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2376                                       ;
2377 016532                              TSTEND: RFLAGS  R0
     016532  104421                                TRAP    C$RFLA
2378 016534  030027  020000                       BIT     R0,#IER
2379 016540  001412                                BEQ     1$              ; BR IF "IER" NOT SET.
2380 016542                                        PRINTF  #ESUM,ERRK      ; PRINT ERROR COUNT.
     016542  013746  016570'                       MOV     ERRK,-(SP)
     016546  012746  016572'                       MOV     #ESUM,-(SP)
     016552  012746  000002                       MOV     #2,-(SP)
     016556  010600                                MOV     SP,R0
     016560  104417                                TRAP    C$PNTF
```

```
                016562  062706  000006                  ADD     #6,SP
     2381       016566  000207                  1$:     RTS     PC
     2382
     2383       016570  000000                  ERRK:   0                       ; LOCAL ERROR COUNT.
     2384       016572     045     101     040  ESUM:   .ASCIZ  /%P %D%A ERRORS/
     2385       016611     105     122     122  EMAXDU: .ASCIZ  /ERROR LIMIT REACHED -- DROPPING UNIT/
     2386                                          .EVEN
     2387
     2388                                          .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
     2389                                  ;+
     2390                                  ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
     2391                                  ;-
     2392       016656  005237  016570'      INCERK: INC     ERRK            ; INCREMENT LOCAL ERROR COUNT
     2393       016662  010046                  MOV     R0,-(SP)        ; SAVE R0
     2394       016664  013700  002200'          MOV     UNITN,R0        ; GET UNIT NUMBER,
     2395       016670  006300                  ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
     2396       016672  062700  003174'          ADD     #ERTABL,R0      ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
     2397       016676  005210                  INC     (R0)            ; INCREMENT THE DEVICE ERROR COUNT
     2398       016700  032710  007777          BIT     #7777,(R0)      ; DID WE OVERFLOW THE FIELD?
     2399       016704  001001                  BNE     1$              ; BR IF NO.
     2400       016706  005310                  DEC     (R0)            ; YES -- BACK IT UP TO 7777.
     2401       016710  012600              1$:     MOV     (SP)+,R0        ; RESTORE R0
     2402       016712  000207                  RTS     PC              ; RETURN TO CALLER.
     2403
     2404       016714  010046              CKEMAX: MOV     R0,-(SP)        ; SAVE R0
     2405       016716  013700  002200'          MOV     UNITN,R0        ; GET UNIT NUMBER
     2406       016722  006300                  ASL     R0              ; ... AND MAKE IT A WORD OFFSET
     2407       016724  016000  003174'          MOV     ERTABL(R0),R0   ; GET ERROR TABLE ENTRY
     2408       016730  042700  170000          BIC     #170000,R0      ; EXTRACT ERROR COUNT FIELD
     2409       016734  020037  002172'          CMP     R0,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
     2410       016740  103004                  BHIS    1$              ; BR IF YES
     2411       016742  023737  016570' 002170'      CMP     ERRK,LERRMAX    ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
     2412       016750  103417                  BLO     2$              ; BR IF NO
     2413       016752                      1$:     RFLAGS  R0              ; GET OPERATOR FLAGS
                016752  104421                  TRAP    C$RFLA
     2414       016754  032700  000040          BIT     #IDU,R0         ; IS DROPPING INHIBITED?
     2415       016760  001013                  BNE     2$              ; BR IF YES.
     2416       016762  012737  177777  003110'      MOV     #-1,DUFLG       ; NO -- DROP THE UNIT
     2417       016770                          ERRDF   4,EMAXDU
                016770  104455                  TRAP    C$ERDF
                016772  000004                  .WORD   4
                016774  016611'                 .WORD   EMAXDU
                016776  000000                  .WORD   0
     2418       017000                          DODU    UNITN
                017000  013700  002200'          MOV     UNITN,R0
                017004  104451                  TRAP    C$DODU
     2419       017006                          DOCLN
                017006  104444                  TRAP    C$DCLN
     2420       017010  012600              2$:     MOV     (SP)+,R0        ; RESTORE R0
     2421       017012  000207                  RTS     PC              ; RETURN TO CALLER
     2422
     2423                                          .SBTTL  CKDROP  - CHECK IF UNIT SHOULD BE DROPPED
     2424                                  ;+
     2425                                  ; CHECK IF UNIT SHOULD BE DROPPED
     2426                                  ;-
     2427       017014  010046              CKDROP: MOV     R0,-(SP)
     2428       017016                          FORCERROR       1$,NOTSSR
```

```
2429 017026                                    RFLAGS  R0
     017026  104421                            TRAP    C$RFLA
2430 017030  032700  000040                    BIT     #IDU,R0
2431 017034  001010                            BNE     1$
2432 017036  011600                            MOV     (SP),R0
2433 017040  012737  177777  003110'           MOV     #-1,DUFLG
2434 017046                                    DODU    UNITN
     017046  013700  002200'                    MOV     UNITN,R0
     017052  104451                            TRAP    C$DODU
2435 017054                                    DOCLN                   ;ABORT THE PASS
     017054  104444                            TRAP    C$DCLN
2436 017056  012600                    1$:     MOV     (SP)+,R0
2437 017060  000207                            RTS     PC
2438
2439                                           .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
2440                                   ;
2441                                   ; SUBROUTINE - DETERMINE CONFIGURATION OF TSU05 SYSTEM.
2442                                   ;
2443 017062                            CONFIG:
2444 017062  004737  015604'                    JSR     PC,SOFINIT
2445 017066  000207                            RTS     PC
2446
2447                                           .SBTTL  KTON,KTOFF      - ENABLE/DISABLE MEMORY MANAGEMENT
2448                                   ;
2449                                   ; SUBROUTINE - ENABLE MEM MGT.
2450                                   ;
2451 017070  005737  003130'           KTON:   TST     KTFLG           ; GOT KT?
2452 017074  001403                            BEQ     1$              ; NO.
2453 017076  012737  000001  177572            MOV     #1,SR0          ; YES. ENABLE KT11.
2454 017104  000207                    1$:     RTS     PC
2455
2456
2457
2458                                   ;
2459                                   ; SUBROUTINE - DISABLE MEM MGT.
2460                                   ;
2461 017106  005737  003130'           KTOFF:  TST     KTFLG           ; GOT KT11?
2462 017112  001405                            BEQ     1$              ; NO.
2463 017114  000240                            NOP
2464 017116  000240                            NOP
2465 017120  012737  000000  177572            MOV     #0,SR0          ; DISABLE KT.
2466 017126  000207                    1$:     RTS     PC
2467
2468                                           .SBTTL  SETMAP  - SETUP PAR6 MAPPING
2469
2470                                   ;+
2471                                   ;
2472                                   ;THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
2473                                   ;AN 22 BIT ADDRESS. THE OFFSET INTO THE PAGE
2474                                   ;IS RETURNED BIASED TO PAR6.
2475                                   ;
2476                                   ;INPUTS:
2477                                   ;
2478                                   ;       R0      HIGH ORDER ADDRESS BITS
2479                                   ;       R1      LOW ORDER ADDRESS BITS
2480                                   ;
2481                                   ;OUTPUTS:
```

```
                        ;
                        ;       R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
                        ;       CARRY   SET IF SUCCESS
                        ;               CLR IF ERROR
                        ;-
017130                  SETMAP:
017130                          SAVREG                          ;SAVE R1-R4 UNTIL NEXT RETURN
017134  005737  003130'          TST     KTFLG                   ;SYSTEM HAVE ABOVE 28K?
017140  001433                  BEQ     10$                     ;BR IF NO
017142  010102                  MOV     R1,R2                   ;SAVE LOW ORDER BITS
        000006                  .REPT   6
                                ASR     R0                      ;CONVERT WORD ADDRESS TO 32W BLOCKS
                                ROR     R1                      ;MAKE IT DOUBLE PRECISION
                                .ENDR
017174  042701  000177          BIC     #177,R1                 ;ALINE FOR LOWER 4K BOUNDARY
017200  020137  003130'          CMP     R1,KTFLG                ;HIGHER THAN EXISTING MEMORY?
017204  103011                  BHIS    10$                     ;BR IF YES
017206  010137  172354          MOV     R1,@#KIPAR6             ;SETUP MAPPING REGISTER PAR6
017212  042702  160000          BIC     #160000,R2              ;SETUP DISPLACEMENT IN PAGE
017216  062702  140000          ADD     #140000,R2              ;ADD IN PAR6 BIAS
017222  010200                  MOV     R2,R0                   ;RETURN IN R0
017224  000261                  SEC                             ;SET SUCCESS
017226  000401                  BR      15$                     ;
017230  000241          10$:    CLC                             ;SET FAILURE
017232  000207          15$:    RTS     PC                      ;RETURN
```

```
                                .SBTTL  FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
                        ;+
                        ; FILL MEMORY WITH A BACKGROUND PATTERN
                        ;
                        ; INPUTS:
                        ;
                        ;       R0 = BACKGROUND PATTERN
                        ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                        ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                        ;
                        ; OUTPUTS:
                        ;
                        ;       NONE
                        ;-
                        ;
017234                  FILLMEM:
017234                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
017240  004737  017106'          JSR     PC,KTOFF                ;DISABLE KT.
017244  010003                  MOV     R0,R3                   ;COPY TEST PATTERN
017246  013701  003122'          MOV     FREE,R1                 ;GET FIRST FREE LOCATION
017252  013702  003124'          MOV     FRESIZ,R2               ;SIZE OF FREE SPACE BELOW 28K.
017256  010321          10$:    MOV     R3,(R1)+                ;STORE A BACKGROUND WORD
017260  005302                  DEC     R2                      ;DONE ALL MEMORY IN FREE SPACE?
017262  003375                  BGT     10$                     ;BR IF NO
017264  005737  003130'          TST     KTFLG                   ; GOT KT?
017270  001502                  BEQ     55$                     ; NO. GET OUT.
017272  004737  017070'          JSR     PC,KTON                 ; YES. ENABLE KT.
017276  005000                  CLR     R0                      ;HIGH ORDER ADDRESS START
017300  013701  003150'          MOV     PST32W,R1               ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                  .REPT   6
```

```
2539                                            CLC             ;CLEAR C BIT
2540                                            ROL     R1      ;CONVERT BLOCKS TO WORDS
2541                                            ROL     R0      ;MAKE IT DOUBLE PRECISION
2542                                            .ENDR
2543 017350  004737  017130'                    JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
2544 017354  010320                     30$:    MOV     R3,(R0)+        ;STORE TEST PATTERN IN >28K ADDRESS
2545 017356  020027  160000                     CMP     R0,#160000      ;END OF PAR6 MAPPING AREA?
2546 017362  103774                             BLO     30$             ;BR IF NO
2547 017364  162700  020000                     SUB     #20000,R0       ;BACKUP INTO PAR6 MAPPING BEGIN
2548 017370  062737  000200  172354             ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
2549 017376  013705  003130'                     MOV     KTFLG,R5        ;GET VALUE FROM MEMORY SIZER
2550 017402  042705  170000                     BIC     #170000,R5      ;ONLY 18 BITS PASS
2551 017406  023705  172354                     CMP     @#KIPAR6,R5     ;END OF MEMORY?
2552 017412  001427                             BEQ     50$             ;BR IF YES
2553 017414  005737  003142'                     TST     T23A            ;PROCESSOR TYPE A
2554 017420  001407                             BEQ     35$             ;NO KEEP GOING
2555 017422  013704  177572                     MOV     SR0,R4          ;GET SR0 CONTENTS
2556 017426  042704  177761                     BIC     #177761,R4      ;CLEAR ALL BUT PAGE NUMBER
2557 017432  022704  000016                     CMP     #16,R4          ;SEE IF PAGE 7
2558 017436  001415                             BEQ     50$             ;EXIT IF THERE
2559 017440  005737  003144'             35$:    TST     T23B            ;PROCESSOR TYPE B
2560 017444  001410                             BEQ     45$             ;NO KEEP GOING
2561 017446  023727  172354  007600             CMP     @#KIPAR6,#7600  ;REACHED 18 BITS?
2562 017454  103001                             BHIS    40$             ;YES
2563 017456  000403                             BR      45$             ;NO KEEP GOING
2564 017460  012737  000020  172516     40$:    MOV     #20,SR3         ;SET MMU RELOCATION
2565 017466  000137  017354'             45$:    JMP     30$             ;KEEP GOING ON ETC.
2566 017472  004737  017106'             50$:    JSR     PC,KTOFF        ; DISABLE KT.
2567 017476  000207                     55$:    RTS     PC
2568
2569                                            .SBTTL  CMPMEM  - COMPARE MEMORY TO BACKGROUND PATTERN
2570                                    ;+
2571                                    ; COMPARE MEMORY WITH A BACKGROUND PATTERN
2572                                    ;
2573                                    ; INPUTS:
2574                                    ;
2575                                    ;       R0 = BACKGROUND PATTERN
2576                                    ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2577                                    ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2578                                    ;
2579                                    ; OUTPUTS:
2580                                    ;
2581                                    ;       CARRY   - SET IF NO ERROR
2582                                    ;       CARRY   - CLR IF ERROR
2583                                    ;
2584                                    ; IMPLICIT OUTPUTS:
2585                                    ;
2586                                    ;       ERRHI   - ERROR HIGH ADDRESS
2587                                    ;       ERRLO   - ERROR LOW ADDRESS
2588                                    ;       EXPD    - EXPECTED DATA
2589                                    ;       RECV    - RECEIVED DATA
2590                                    ;-
2591 017500                             CMPMEM:
2592 017500                                     SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
2593 017504  010003                             MOV     R0,R3           ;COPY TEST PATTERN
2594 017506  004737  017106'                     JSR     PC,KTOFF        ;DISABLE KT.
2595 017512  013701  003122'                     MOV     FREE,R1         ;GET FIRST FREE LOCATION
```

```
2596 017516  013702  003124'                  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
2597 017522  020311                    10$:    CMP     R3,(R1)         ;FREE SPACE LOCATION EQUAL TO EXPD?
2598 017524  001411                            BEQ     15$             ;BR IF YES
2599 017526  010137  002236'                  MOV     R1,ERRLO        ;SAVE ADDRESS IN ERROR
2600 017532  005037  002234'                  CLR     ERRHI           ;NO HIGH ADDRESS
2601 017536  010337  002230'                  MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
2602 017542  011137  002232'                  MOV     (R1),RECV       ;SAVE RECV FOR ERROR REPORT
2603 017546  000474                            BR      50$             ;
2604 017550  005721                    15$:    TST     (R1)+           ;POINT TO NEXT ADDRESS
2605 017552  005302                            DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
2606 017554  003362                            BGT     10$             ;BR IF NO
2607 017556  005737  003130'                  TST     KTFLG           ; GOT KT?
2608 017562  001472                            BEQ     55$             ; NO, GET OUT.
2609 017564  004737  017070'                  JSR     PC,KTON         ; YES. ENABLE KT.
2610 017570  005000                            CLR     R0              ;HIGH ORDER ADDRESS START
2611 017572  013701  003150'                  MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
2612         000006                            .REPT   6
2613                                           ROL     R1              ;CONVERT BLOCKS TO WORDS
2614                                           ROL     R0              ;MAKE IT DOUBLE PRECISION
2615                                           .ENDR
2616 017626  042701  000177                    BIC     #177,R1         ;ALINE 4K BOUNDARY
2617 017632  010046                            MOV     R0,-(SP)        ;SAVE HIGH ORDER
2618 017634  010146                            MOV     R1,-(SP)        ;SAVE LOW ORDER
2619 017636  004737  017130'                  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
2620 017642  010004                            MOV     R0,R4           ;COPY ADDRESS BIASED TO PAR6
2621 017644  012601                            MOV     (SP)+,R1        ;RESTORE LOW ORDER IN NON PAR6 FORMAT
2622 017646  012600                            MOV     (SP)+,R0        ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
2623 017650  020314                    30$:    CMP     R3,(R4)         ;ABOVE 28K LOCATION EQUAL EXPD?
2624 017652  001411                            BEQ     32$             ;BR IF YES
2625 017654  010037  002234'                  MOV     R0,ERRHI        ;SAVE HIGH ORDER IN ERROR
2626 017660  010137  002236'                  MOV     R1,ERRLO        ;SAVE LOW ORDER IN ERROR
2627 017664  010337  002230'                  MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
2628 017670  011437  002232'                  MOV     (R4),RECV       ;SAVE RECV FOR ERROR REPORT
2629 017674  000421                            BR      50$             ;
2630 017676  062701  000002            32$:    ADD     #2,R1           ;UPDATE NON PAR6 ADDRESS
2631 017702  005500                            ADC     R0              ;MAKE IT DOUBLE PRECISION ADD
2632 017704  062704  000002                    ADD     #2,R4           ;UPDATE PAR FORMAT ADDRESS
2633 017710  020427  160000                    CMP     R4,#160000      ;END OF PAR6 MAPPING AREA?
2634 017714  103755                            BLO     30$             ;BR IF NO
2635 017716  162704  020000                    SUB     #20000,R4       ;BACKUP INTO PAR6 MAPPING BEGIN
2636 017722  062737  000200  172354            ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
2637 017730  023737  172354  003130'           CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
2638 017736  101744                            BLOS    30$             ;BR IF NO
2639 017740  004737  017106'            50$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2640 017744  000241                            CLC                     ;SET FAILURE
2641 017746  000403                            BR      60$             ;
2642 017750  004737  017106'            55$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2643 017754  000261                            SEC                     ;SET SUCCESS
2644 017756  000207                    60$:    RTS     PC
2645
2646                                           .SBTTL  REGSAV    SAVE R1-R5 ON STACK
2647                                   ;+
2648                                   ;
2649                                   ;ROUTINE TO
2650                                   ;SAVE R1 THROUGH R5 ON THE STACK
2651                                   ;
2652                                   ;CALLING SEQUENCE:
```

```
                                        ;
                                        ;       JSR     R5,REGSAV
                                        ;
                                        ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
                                        ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
                                        ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
                                        ;REGISTERS.
                                        ;
                                        ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
                                        ;CALLED VIA A JSR PC INSTRUCTION
                                        ;
                                        ;-

017760                                  REGSAV:
017760  010446                                  MOV     R4,-(SP)
017762  010346                                  MOV     R3,-(SP)
017764  010246                                  MOV     R2,-(SP)
017766  010146                                  MOV     R1,-(SP)
017770  010546                                  MOV     R5,-(SP)
017772  016605  000012                          MOV     10.(SP),R5
017776  004736                                  JSR     PC,@(SP)+
020000  012601                                  MOV     (SP)+,R1
020002  012602                                  MOV     (SP)+,R2
020004  012603                                  MOV     (SP)+,R3
020006  012604                                  MOV     (SP)+,R4
020010  012605                                  MOV     (SP)+,R5
020012  000207                                  RTS     PC

                                        .SBTTL  GETPAT  - GET 8 BIT PATTERN FROM OPERATOR
                                        ;+
                                        ;
                                        ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       NONE.
                                        ;
                                        ;OUTPUTS:
                                        ;
                                        ;       R0      OCTAL NUMBER FROM THE OPERATOR
                                        ;
                                        ;CALLING SEQUENCE:
                                        ;
                                        ;       JSR     PC,GETPAT
                                        ;
                                        ;-

020014                                  GETPAT::
020014                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
020020                                  1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
020020  104443                                  TRAP    C$GMAN
020022  000406                                  BR      10000$
020024  020050'                                 .WORD   PATDAT
020026  000022                                  .WORD   T$CODE
020030  020052'                                 .WORD   DATASC
020032  000377                                  .WORD   377
020034  000000                                  .WORD   T$LOLIM
```

```
                020036  000377                          .WORD   T1HILIM
                020040                          10000$:
     2703       020040                                  BNCOMPLETE      1$      ;RETRY IF ERROR
                020040  103767                          BCC     1$
     2704       020042  013700  020050'                MOV     PATDAT,R0       ;DATA PATTERN FROM OPERATOR
     2705       020046  000207                          RTS     PC              ;RETURN TO CALLER
     2706
     2707                                               ;*
     2708                                               ;LOCAL DATA AREA
     2709                                               ;-
     2710
     2711       020050  000000                          PATDAT: .WORD   0               ;TEMPORARY STORAGE FOR DATA
     2712       020052     105     116     124         DATASC: .ASCIZ  'ENTER DATA PATTERN'
     2713                                                       .EVEN
     2714
     2715                                                       .SBTTL  GETSEL  - ISSUE MENU AND GET OPERATOR RESPONSE
     2716                                               ;*
     2717                                               ;
     2718                                               ;ROUTINE TO ISSUE A MENU AND GET
     2719                                               ;THE OPERATOR'S RESPONSE.
     2720                                               ;
     2721                                               ;INPUTS:
     2722                                               ;
     2723                                               ;       R0      ADDRESS OF ASCIZ STRING OF MENU
     2724                                               ;       R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
     2725                                               ;
     2726                                               ;OUTPUTS:
     2727                                               ;
     2728                                               ;       R0      NUMBER OF THE OPERATOR'S SELECTION
     2729                                               ;
     2730                                               ;-
     2731
     2732       020076                          GETSEL::
     2733       020076                                  SAVREG                          ;SAVE GENERAL REGISTERS
     2734       020102  010002                          MOV     R0,R2                   ;SAVE THE MENU ADDRESS
     2735       020104  010203                  1$:     MOV     R2,R3                   ;START OF MENU STRING
     2736       020106  005713                  2$:     TST     (R3)                    ;END OF ASCII ?
     2737       020110  001412                          BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
     2738       020112                                  PRINTF  @SELASC,(R3)+           ;DISPLAY THE MENU
                020112  012346                          MOV     (R3)+,-(SP)
                020114  012746  020262'                MOV     @SELASC,-(SP)
                020120  012746  000002                  MOV     #2,-(SP)
                020124  010600                          MOV     SP,R0
                020126  104417                          TRAP    C1PNTF
                020130  062706  000006                  ADD     #6,SP
     2739       020134  000764                          BR      2$
     2740       020136                          3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
                020136  104443                          TRAP    C1GMAN
                020140  000406                          BR      10001$
                020142  020316'                         .WORD   MENRES
                020144  000042                          .WORD   T1CODE
                020146  020267'                         .WORD   MENASC
                020150  177777                          .WORD   -1
                020152  000000                          .WORD   T1LOLIM
                020154  177777                          .WORD   T1HILIM
                020156                          10001$:
     2741       020156                                  BNCOMPLETE      1$      ;RETRY IF ERROR
```

```
                020156  103352                          BCC     1$
2742            020160  013700  020316'                 MOV     MENRES,R0       ;GET THE OPERATOR'S REPLY
2743            020164  020001                          CMP     R0,R1           ;COMPARE TO MAXIMUM ALLOWED
2744            020166  101411                          BLOS    5$              ;BRANCH IF OK
2745            020170                                  PRINTF  #MENERR         ;DISPLAY ERROR MESSAGE
                020170  012746  020214'                 MOV     #MENERR,-(SP)
                020174  012746  000001                  MOV     #1,-(SP)
                020200  010600                          MOV     SP,R0
                020202  104417                          TRAP    C$PNTF
                020204  062706  000004                  ADD     #4,SP
2746            020210  000735                          BR      1$              ;RETRY
2747            020212  000207                  5$:     RTS     PC              ;RETURN TO CALLER
2748            020214     045     116     045  MENERR: .ASCIZ  '%N%A *** Menu Selection Too Large ***'
2749            020262     045     116     045  SELASC: .ASCIZ  '%N%T'
2750            020267     105     156     164  MENASC: .ASCIZ  'Enter Menu Selection: '
2751                                                    .EVEN
2752            020316  000000                  MENRES:         .WORD   0
2753
2754                                                    .SBTTL  CHKMAN  - CHECK MANUAL INTERVENTION LEGALITY
2755                                            ;+
2756                                            ;
2757                                            ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2758                                            ;
2759                                            ;INPUT:
2760                                            ;
2761                                            ;       NONE.
2762                                            ;
2763                                            ;OUTPUT:
2764                                            ;
2765                                            ;       CARRY   0       MANUAL INTERVENTION NOT ALLOWED
2766                                            ;               1       MANUAL INTERVENTION IS OK
2767                                            ;
2768                                            ;SIDE EFFECTS:
2769                                            ;
2770                                            ;       A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2771                                            ;       NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2772                                            ;       ALLOWED.
2773                                            ;
2774                                            ;-
2775
2776            020320                          CHKMAN::
2777            020320                                  SAVREG                  ;SAVE THE REGISTERS
2778            020324                                  MANUAL                  ;SEE IF MANUAL INTERVENTION OK
                020324  104450                          TRAP    C$MANI
2779            020326                                  BCOMPLETE 1$            ;BRANCH IF ALLOWED
                020326  103411                          BCS     1$
2780            020330                                  PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
                020330  012746  020354'                 MOV     #NOMAN,-(SP)
                020334  012746  000001                  MOV     #1,-(SP)
                020340  010600                          MOV     SP,R0
                020342  104417                          TRAP    C$PNTF
                020344  062706  000004                  ADD     #4,SP
2781            020350  000241                          CLC                     ;CLEAR CARRY FOR ERROR
2782            020352  000207                  1$:     RTS     PC              ;RETURN
2783
2784            020354     045     116     045  NOMAN:  .ASCIZ  '%N%A *** Manual Intervention not Allowed - Test Aborted ***'
2785                                                    .even
```

```
                                ;       SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
                                ;
                                ;
                                ;INPUTS:
                                ;OUTPUTS:
                                ;       The NXMFLG is set if we can test.
                                ;       The NXMLO and NXMHI addresses are setup.
                                ;-

021026                          MEMCK::

021026                                  SAVREG                          ;SAVE THE REGISTERS
021032  005037  003134'                  CLR     NXMFLG                  ;CLEAR THE FLAG
021036  005037  003136'                  CLR     NXMLO                   ;CLEAR THE TEST ADDRESS LO
021042  005037  003140'                  CLR     NXMHI                   ;CLEAR THE TEST ADDRESS HI
021046  032737  170000  002120'          BIT     #170000,L$HIME          ;CHECK FOR MORE THAN 18 BITS INDICATED
                                                                        ;FROM THE SUPERVISOR
021054  001050                          BNE     14$                     ;BR, IF MAP BOX ETC.
021056  005737  003144'                  TST     T23B                    ;IS IT A PROCESSOR TYPE B?
021062  001407                          BEQ     1$                      ;NO
021064  023727  002120'  007777          CMP     L$HIME,#7777            ; GREATER THAN 128K
021072  103406                          BLO     2$                      ; NO
021074  004737  021222'                  JSR     PC,NXMTST               ;SETUP THE ADDRESS
021100  000427                          BR      13$                     ;SET THE FLAG AND EXIT
021102  005737  003142'          1$:     TST     T23A                    ;IS IT A PROCESSOR TYPE A?
021106  001413                          BEQ     4$                      ;NO
021110  023727  002120'  003777  2$:     CMP     L$HIME,#5777            ;GREATER THAN 96K
021116  101027                          BHI     14$                     ;YES,23A/23B WITH 128K MEMORY
021120  023727  002120'  003777          CMP     L$HIME,#3777            ;GREATER THAN 64K BUT LESS THAN 92K?
021126  103403                          BLO     4$                      ;NO, CHECK 24K
021130  004737  021222'                  JSR     PC,NXMTST               ;SETUP THE ADDRESS
021134  000411                          BR      13$                     ;SET THE FLAG AND EXIT
021136  023727  002120'  001577  4$:     CMP     L$HIME,#1577            ;GREATER THAN 24K BUT LESS THAN 64K?
021144  103414                          BLO     14$                     ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
021146  004737  021222'                  JSR     PC,NXMTST               ;SETUP THE ADDRESS
021152  062737  000077  003140'          ADD     #77,NXMHI               ;FOOL THE 11/02 & 11/03
021160  032737  177774  003140'  13$:     BIT     #177774,NXMHI           ;ANY MORE THAN 18 BITS SET?
021166  001014                          BNE     15$                     ;BR, IF MORE THAN 18 BITS SET
021170  005237  003134'                  INC     NXMFLG                  ;SET THE FLAG
021174  000411                          BR      15$                     ;EXIT
021176  000410                  14$:     BR      15$                     ;NOP FOR PRINTOUT
021200                                  PRINTF  #NONMEM                 ;TELL THEM & EXIT ***NO PRINT*****
021200  012746  005452'                  MOV     #NONMEM,-(SP)
021204  012746  000001                  MOV     #1,(SP)
021210  010600                          MOV     SP,R0
021212  104417                          TRAP    C%PRTF
021214  062706  000004                  ADD     #4,SP
021220  000207                  15$:     RTS     PC                      ;RETURN


                                ;+
                                ;       SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
                                ;
                                ;OUTPUTS:NXMLO,NXMHI            ;SETUP WITH NXM ADDRESS
                                ;
                                ;-
```

```
2953 021222  013701  002120'        NXMTST: MOV     L$HIME,R1       ;GET TOP OF MEMORY
2954 021226  062701  000200                 ADD     #200,R1         ;MAKE IT I/O BLOCK OR OTHER NXM
2955 021232  042701  000177                 BIC     #177,R1
2956 021236  010102                        MOV     R1,R2           ;RESAVE RESULTS
2957         000006                        .REPT   6
2958                                       ASL     R1              ;PUT IN PLACE FOR XFER
2959                                       .ENDR
2960 021254  010137  003136'               MOV     R1,NXMLO        ;SAVE TEST ADDRESS LOW
2961         000012                        .REPT   10.
2962                                       ASR     R2              ;PUT IN PLACE FOR XFER
2963                                       .ENDR
2964 021304  042702  177700                BIC     #177700,R2      ;DON'T WANT ILA!
2965 021310  010237  003140'               MOV     R2,NXMHI        ;SAVE TEST ADDRESS HIGH
2966 021314  000207                        RTS     PC              ;RETURN
2967
2968
2969
2970
2971 021316                                ENDMOD
```

```
                                        .TITLE  TSV4 - MISCELLANEOUS SECTIONS

021316                                  BGNMOD  TSV4
021316                          TSV4::


                                        .SBTTL  PROTECTION TABLE
021316                                  BGNPROT
021316                          L$PROT::
021316  177777  177777  177777        .WORD   -1, -1, -1, -1          ;NO DEVICE PROTECTION REQUIRED.
021326                                  ENDPROT

                                        .SBTTL  INITIALIZE SECTION

                                ;++
                                ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;AT THE BEGINNING OF EACH PASS.
                                ;
                                ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
                                ;IF "CONTINUE", NOTHING IS REQUIRED.
                                ;
                                ;--
                                ;+
                                ;INSERT TEMPORARY JUMP TO ODT
                                ;-
021326                                  BCNINIT
021326                          L$INIT::
021326  005037  002224'        40$:    CLR     EXTFEA
021332  005037  003134'                CLR     NXMFLG
021336  012737  006170'  002176'       MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021344  005037  003152'                CLR     SIFLAG                  ;CLEAR "SOFT INJT" FLAG
021350  005037  003132'                CLR     KTENABLE                ;CLEAR TEST ABOVE 28K FLAG
021354  005037  002300'                CLR     RAMSIZ                  ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021360                                  READEF  #EF.CONTINUE
021360  012700  000036                MOV     #EF.CONTINUE,R0
021364  104447                        TRAP    C$REFG
021366                                  BNCOMPLETE 1$
021366  103023                        BCC     1$
021370  023737  002200'  002012'       CMP     UNITN,L$UNIT            ;UNIT IN RANGE?
021376  103070                        BHIS    4$                      ;BR IF NO.
021400  005737  003110'                TST     DUFLG                   ;DROPPED UNIT?
021404  100472                        BMI     NXTU                    ;BR IF YES
021406  013701  002200'                MOV     UNITN,R1
021412  006301                        ASL     R1
021414  005761  003174'                TST     ERTABL(R1)
021420  001516                        BEQ     SETU
021422  0327 .  040000  003174'        BIT     #BIT14,ERTABL(R1)       ;DROPPED?
021430  001060                        BNE     NXTU
021432                                  EXIT    INIT                    ;DO NOTHING IF "CONTINUE".
021432  104456                        TRAP    C$EXIT
021434  000416                        .WORD   L10030-.
021436                          1$:     READEF  #EF.NEW
021436  012700  000035                MOV     #EF.NEW,R0
021442  104447                        TRAP    C$REFG
021444                                  BNCOMPLETE NXTU                 ;TAKE NEXT UNIT IF NOT NEW PASS.
021444  103052                        BCC     NXTU
021446                                  READEF  #EF.START
```

INITIALIZE SECTION

```
                021446  012700  000040                  MOV     #EF.START,R0
                021452  104447                          TRAP    C$REFG
             57 021454                                  BCOMPLETE 2$
                021454  103404                          BCS     2$
             58 021456                                  READEF  #EF.RESTART
                021456  012700  000037                  MOV     #EF.RESTART,R0
                021462  104447                          TRAP    C$REFG
             59 021464                                  BNCOMPLETE 31$
                021464  103031                          BCC     31$
             60 021466                          2$:                             ;1ST PASS, BUS-INIT...
             61 021466                                  BRESET                  ;BUS RESET.
                021466  104433                          TRAP    C$RESET
             62 021470  005037  002212'                 CLR     TSTCNT          ;NUMBER OF TESTS RUN IN PASS
             63 021474  005037  002220'                 CLR     FATFLG          ;CLEAR FATAL ERROR COUNT
             64 021500  005037  003142'                 CLR     T23A            ;CLEAR PROCESSOR TYPE A FLAG
             65 021504  005037  003144'                 CLR     T23B            ;CLEAR PROCESSOR TYPE B FLAG
             66                                 ;       MOV     #340,-(SP)
             67                                 ;       MOV     #20$,-(SP)      ;RETURN TO DEBUGGER
             68                                 ;       JMP     0.ODT           ;;@ENTER THE DEBUGGER
             69 021510  005037  003376'                 CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
             70 021514                          20$:
             71 021514  012737  177777  002202'        MOV     #-1,QVP                 ;...QUICK VERIFY...
             72 021522  004737  020450'                JSR     PC,ENVIRN               ;SET ENVIRONMENT.
             73 021526  004737  020536'                JSR     PC,KTINIT               ;INITIALIZE KT MEMORY MANAGEMENT
             74 021532  012700  003174'                MOV     #ERTABL,R0
             75 021536  005020                 30$:    CLR     (R0)+                   ;CLEAR THE ERROR TABLE
             76 021540  020027  003374'                CMP     R0,#ERTABE
             77 021544  103774                         BLO     30$
             78 021546  000404                         BR      4$
             79 021550  005037  002202'        31$:    CLR     QVP
             80 021554  000137  021624'                JMP     PASRPT                  ;GO REPORT THE STATUS
             81
             82 021560                          4$:
             83 021560  012737  177777  002200' NEWPAS: MOV     #-1,UNITN               ;INIT UNIT NUMBER...
             84 021566  005037  002216'                CLR     DEVCNT                  ;CLEAR COUNT OF DEVICES RUNNING
             85 021572                          NXTU:   BREAK
                021572  104422                          TRAP    C$BRK
             86 021574  005237  002200'                INC     UNITN                   ;...AND SET NEXT UNIT NUMBER.
             87 021600  023737  002200'  002012'       CMP     UNITN,L$UNIT
             88 021606  103423                         BLO     SETU
             89 021610  012737  177777  003110'        MOV     #-1,DUFLG
             90 021616  000401                         BR      11$
             91 021620                                  DOCLN                   ;ABORT. NO MORE UNITS.
                021620  104444                          TRAP    C$DCLN
             92 021622  000240                 11$:    NOP
             93 021624                          PASRPT:
             94 021624  023727  002012'  000001        CMP     L$UNIT,#1               ;HOW MANY UNITS SELECTED?
             95 021632  101752                         BLOS    NEWPAS                  ;BR IF ONLY 1
             96 021634  005737  002216'                TST     DEVCNT                  ;ARE ANY STILL RUNNING?
             97 021640  001747                         BEQ     NEWPAS                  ;BR IF NO
             98 021642                                  RFLAGS  R0
                021642  104421                          TRAP    C$RFLA
             99 021644  032700  000100                 BIT     #ISR,R0                 ;SHOULD WE PRINT STATISTICS
            100 021650  001343                         BNE     NEWPAS                  ;BR IF NO
            101
            102 021652                                  DORPT
                021652  104424                          TRAP    C$DRPT
```

```
                        BR      NEWPAS
                10$:
                SETU:   GPHARD  UNITN,R0                ;GET UNIT N P-TABLE POINTER.
021656  013700  002200'        MOV     UNITN,R0
021662  104442                TRAP    C$GPHRD
                        BNCOMPLETE NXTU                 ;BR IF UNIT NOT AVAILABLE.
021664  103342                BCC     NXTU
021666  005037  003110'        CLR     DUFLG           ;CLEAR "DROPPED" FLAG.
021672  005237  002216'        INC     DEVCNT
021676  012001                MOV     (R0)+,R1        ;GET 1ST REGISTER ADDRESS.
021700  010137  002204'        MOV     R1,CSRADDR      ;ADDRESS OF REGISTERS OF UNIT UNDER TEST

021704  012001                MOV     (R0)+,R1        ;GET VECTOR ADDRESS.
                        ;MOV    (R0),R2         ;GET INTERRUPT PRIORITY
                        ;MOV    R2,IPRI         ;SET INTERRUPT PRIORITY.
021706  010137  002206'        MOV     R1,IVEC         ;SET INTERRUPT VECTOR POINTER...
021712  012721  016026'        MOV     #INTR,(R1)+     ;...VECTOR...
021716  013721  002210'        MOV     IPRI,(R1)+      ;...AND PRIORITY.

021722                  1$:
                        ;       TST     QVP             ;1ST PASS ??
                        ;       BEQ     5$              ;NO, SKIP THE PASS 1 STUFF.


                        ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
                        ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
                        ;
021722  013701  002200'        MOV     UNITN,R1
021726  006301                ASL     R1
021730  052761  100000  003174'    BIS     #BIT15,ERTABL(R1)       ;SAY DEVICE RUNNING
021736  005037  005602'        CLR     EXTA                    ;CLEAR ERROR EXTENSION FLAG.
021742  023727  002012' 000001    CMP     L$UNIT,#1               ;ARE WE TESTING MULTIPLE UNITS?
021750  101416                BLOS    10$                     ;BR IF NO.
021752                  RFLAGS  R0                      ;YES -- GET OPERATOR FLAGS.
021752  104421                TRAP    C$RFLA
021754  032700  001000        BIT     #PNT,R0                 ;SHOULD WE PRINT UNIT #?
021760  001412                BEQ     10$                     ;BR IF NOT.
021762                  PRINTF  #PUNIT,UNITN            ;PRINT THE UNIT #
021762  013746  002200'        MOV     UNITN,-(SP)
021766  012746  022054'        MOV     #PUNIT,-(SP)
021772  012746  000002        MOV     #2,-(SP)
021776  010600                MOV     SP,R0
022000  104417                TRAP    C$PNTF
022002  062706  000006        ADD     #6,SP
022006                  10$:
022006  005037  003112'        CLR     NODEV
022012  013701  002204'        MOV     CSRADDR,R1              ;ADDRESS OF FIRST REGISTER
022016  010102                MOV     R1,R2                   ;START OF REGISTERS
022020  062702  000002        ADD     #TSSR,R2                ;ADDRESS OF TSSR REGISTER
022024  004737  016206'        JSR     PC,XNXM                 ;TEST BOTH CONTROLLER REGISTERS...
022030  103005                BCC     2$                      ;...AND BR IF ALL OK.
022032  010137  003112'        MOV     R1,NODEV                ;FLAG DEVICE AS NON-EXISTENT
022036  012737  177777  003110'    MOV     #-1,DUFLG               ;DROP THIS UNIT.
022044                  2$:
                        ;
                        ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
```

```
                                        ;
022044                          5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
022044  012700  000000                  MOV     #PRI00,R0
022050  104441                          TRAP    C$SPRI
022052                                  ENDINIT
022052                          L10030:
022052  104411                          TRAP    C$INIT

022054    045     116     045   PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
                                        .EVEN

                                        .SBTTL  ADD AND DROP UNITS SECTIONS

                                        ;++
                                        ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                        ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
                                        ;    OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
                                        ;--
022122                                  BGNAU
022122                          L$AU::
022122  010001                          MOV     R0,R1                   ; GET UNIT TO BE ADDED (R0)
022124  006301                          ASL     R1                      ; MAKE IT A WORD INDEX
022126  052761  100000  003174'         BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
022134  042761  040000  003174'         BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
022142                                  PRINTF  #1$,R0
022142  010046                          MOV     R0,-(SP)
022144  012746  022170'                 MOV     #1$,-(SP)
022150  012746  000002                  MOV     #2,-(SP)
022154  010600                          MOV     SP,R0
022156  104417                          TRAP    C$PNTF
022160  062706  000006                  ADD     #6,SP
022164                                  EXIT    AU
022164  000167                          .WORD   J$JMP
022166  000026                          .WORD   L10031-2-.
022170    045     116     045   1$:     .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN

022216                                  ENDAU           ; UNUSED.
022216                          L10031:
022216  104452                          TRAP    C$AU
                                        ;++
                                        ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                        ; TO BE REMOVED FROM THE TEST LIST.
                                        ;
                                        ; SUPVSR DOES THE "DROPPING".  THIS IS JUST TO TELL THE MAN.
                                        ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
                                        ; COMMAND, OTHERWISE REMAIN INACTIVE.  THE 'DISPLAY" COMMAND
                                        ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
                                        ; WHICH ARE STILL ACTIVE.
                                        ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.

022220                                  BGNDU
022220                          L$DU::
022220  012737  177777  003110'         MOV     #-1,DUFLG
022226  010001                          MOV     R0,R1
022230  006301                          ASL     R1
022232  052761  140000  003174'         BIS     #140000,ERTABL(R1)      ; SAY DROPPED
```

```
    191 022240  000240  000240  000240          240,240,240                     ; ?????????
    192 022246                                  PRINTF  #1$,R0
        022246  010046                          MOV     R0,-(SP)
        022250  012746  022274'                 MOV     #1$,-(SP)
        022254  012746  000002                  MOV     #2,-(SP)
        022260  010600                          MOV     SP,R0
        022262  104417                          TRAP    C$PNTF
        022264  062706  000006                  ADD     #6,SP
    193 022270                                  EXIT    DU
        022270  000167                          .WORD   J$JMP
        022272  000030                          .WORD   L10032-2-.
    194 022274     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
    195                                         .EVEN
    196 022324                                  ENDDU
        022324                          L10032:
        022324  104453                          TRAP    C$DU
    197                                 ;++
    198                                 ; AUTO-DROP CODE SECTION.
    199                                 ;--
    200 022326                                  BGNAUTO
        022326                          L$AUTO::
    201 022326  013705  002204'                 MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
    202 022332  012703  000550                  MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
    203 022336  004737  016060'         10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
    204 022342  103420                          BCS     20$                     ;LEAVE WHEN SSR IS SET
    205 022344                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
        022344  012727  000372                  MOV     #250.,(PC)+
        022350  000000                          .WORD   0
        022352  013727  002116'                 MOV     L$DLY,(PC)+
        022356  000000                          .WORD   0
        022360  005367  177772                  DEC     -6(PC)
        022364  001375                          BNE     .-4
        022366  005367  177756                  DEC     -22(PC)
        022372  001367                          BNE     .-20
    206 022374  005303                          DEC     R3                      ;BUMP COUNTER DOWN
    207 022376  001357                          BNE     10$                     ;KEEP GOING
    208 022400  004737  017014'                 JSR     PC,CKDROP               ;TRY AND DROP UNIT
    209 022404                          20$:
    210 022404                                  ENDAUTO                 ; UNUSED.
        022404                          L10033:
        022404  104461                          TRAP    C$AUTO
    211
    212                                         .SBTTL  CLEAN-UP AND REPORT CODING SECTIONS
    213
    214                                 ;++
    215                                 ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
    216                                 ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
    217                                 ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
    218                                 ;--
    219 022406                                  BGNCLN
        022406                          L$CLEAN::
    220 022406  013705  002204'                 MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
    221 022412  005737  003110'                 TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
    222 022416  100405                          BMI     1$              ;...AND GROSS CONTROLLER FAULT...
    223                                                                 ;...DON'T TRY TO XCT CLEANUP CODE.
    224
    225 022420  012765  000000  000002          MOV     #0,TSSR(R5)     ;DO SOFT INIT
```

```
    226 022426  004737  016060'                 JSR     PC,WAITF
    227 022432                              1$:
    228 022432                              2$:     ENDCLN
        022432                              L10034:
        022432  104412                            TRAP    C$CLEAN
    229                                     ;++
    230                                     ; THE REPORT CODING SECTION CONTAINS THE
    231                                     ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
    232                                     ;--
    233 022434                                      BGNRPT
        022434                              L$RPT::
    234 022434                                      PRINTS  #DEVSUM
        022434  012746  022676'                       MOV     #DEVSUM,-(SP)
        022440  012746  000001                        MOV     #1,-(SP)
        022444  010600                        MOV     SP,R0
        022446  104416                        TRAP    C$PNTS
        022450  062706  000004                ADD     #4,SP
    235 022454  010246                        MOV     R2,-(SP)
    236 022456  010346                        MOV     R3,-(SP)
    237 022460  010446                        MOV     R4,-(SP)
    238 022462  012704  003174'               MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
    239 022466  005003                        CLR     R3              ; CLEAR UNIT NUMBER
    240 022470  011402                  1$:   MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
    241 022472  001467                        BEQ     4$              ; ZERO IF UNIT NOT RUN
    242 022474  100066                        BPL     4$
    243 022476  032702  040000               BIT     #BIT14,R2       ; WAS UNIT DROPPED?
    244 022502  001015                        BNE     2$              ; BR IF YES
    245 022504  042702  170000               BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
    246 022510                                      PRINTS  #DEVONL,R3,R2   ; PRINT
        022510  010246                        MOV     R2,-(SP)
        022512  010346                        MOV     R3,-(SP)
        022514  012746  022733'               MOV     #DEVONL,-(SP)
        022520  012746  000003                MOV     #3,-(SP)
        022524  010600                        MOV     SP,R0
        022526  104416                        TRAP    C$PNTS
        022530  062706  000010                ADD     #10,SP
    247 022534  000446                        BR      4$
    248 022536  020227  160000          2$:   CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
    249 022542  001012                        BNE     3$              ; BR IF NO
    250 022544                                      PRINTS  #DEVNXR,R3
        022544  010346                        MOV     R3,-(SP)
        022546  012746  023003'               MOV     #DEVNXR,-(SP)
        022552  012746  000002                MOV     #2,-(SP)
        022556  010600                        MOV     SP,R0
        022560  104416                        TRAP    C$PNTS
        022562  062706  000006                ADD     #6,SP
    251 022566  000431                        BR      4$
    252 022570  020227  160001          3$:   CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
    253 022574  001012                        BNE     30$             ; BR IF NO.
    254 022576                                      PRINTS  #DEVNRD,R3
        022576  010346                        MOV     R3,-(SP)
        022600  012746  023065'               MOV     #DEVNRD,-(SP)
        022604  012746  000002                MOV     #2,-(SP)
        022610  010600                        MOV     SP,R0
        022612  104416                        TRAP    C$PNTS
        022614  062706  000006                ADD     #6,SP
    255 022620  000414                        BR      4$
```

```
256 022622  042702  170000          30$:    BIC     #^C7777,R2
257 022626                                  PRINTS  #DEVDRO,R3,R2
    022626  010246                          MOV     R2,-(SP)
    022630  010346                          MOV     R3,-(SP)
    022632  012746  023146'                MOV     #DEVDRO,-(SP)
    022636  012746  000003                  MOV     #3,-(SP)
    022642  010600                          MOV     SP,R0
    022644  104416                          TRAP    C$PNTS
    022646  062706  000010                  ADD     #10,SP
258 022652  062704  000002          4$:     ADD     #2,R4
259 022656  005203                          INC     R3
260 022660  020427  003374'                CMP     R4,#ERTABE
261 022664  103701                          BLO     1$
262 022666  012604                          MOV     (SP)+,R4
263 022670  012603                          MOV     (SP)+,R3
264 022672  012602                          MOV     (SP)+,R2
265 022674                                  ENDRPT          ; UNUSED.
    022674                          L10035:
    022674  104425                          TRAP    C$RPT
266
267
266 022676     045     116     045  DEVSUM: .ASCIZ  /%N%ADEVICE STATUS SUMMARY:%N/
269 022733     045     101     040  DEVONL: .ASCIZ  /%A  UNIT %D3%A    ONLINE,  ERRORS = %D%N/
270 023003     045     101     040  DEVNXR: .ASCIZ  /%A  UNIT %D3%A    DROPPED, NON-EXISTENT REGISTER%N/
271 023065     045     101     040  DEVNRD: .ASCIZ  /%A  UNIT %D3%A    DROPPED, NOT READY AT STARTUP%N/
272 023146     045     101     040  DEVDRO: .ASCIZ  /%A  UNIT %D3%A    DROPPED, ERRORS = %D%N/
273                                         .EVEN
274
275 023216                                  ENDMOD
276
277
278
```

```
                                        .TITLE  TSV5A - HARDWARE TESTS
                                        BGNMOD  TSV5
023216                          TSV5::
                                        .SBTTL  TEST  1: BUS RESET TEST

                                        THIS TEST VERIFIES THAT THE M7455 MODULE'S DEVICE REGISTERS ARE
                                        ACCESSIBLE ON THE BUS (SUBTEST 1) AND THEN CHECKS THAT THE
                                        BUILT-IN INITIALIZATION SELF-TEST MICRODIAGNOSTIC DID NOT FIND
                                        ANY BASIC PROBLEMS IN THE MODULE.  AREAS OF LOGIC TESTED BY THE
                                        SELF-TEST SEQUENCE ARE AS FOLLOWS:  ROM AND PIPELINE REGISTER,
                                        SEQUENCER, INTERNAL BUSES, 2901 MICROPROCESSOR, AND, RAM.  THIS
                                        TEST INITIALIZES THE CONTROLLER BY ISSUING THE BUS INIT SIGNAL
                                        VIA A RESET INSTRUCTION, OR BY WRITING INTO THE TSSR REGISTER,
                                        WAITS A PERIOD OF TIME (TO ALLOW THE CONTROLLER'S INITIALIZATION
                                        MICRODIAGNOSTIC SEQUENCE TO BE COMPLETED), AND THEN CHECKS THE
                                        CONTENTS OF THE TSSR REGISTER.  SUCCESSFUL INITIALIZATION IS
                                        INDICATED BY SUBSYSTEM READY (SSR) AND NEED BUFFER ADDRESS (NBA)
                                        BITS BEING SET (1) AND ALL OTHER BITS (EXCEPT A17 AND A16 AND
                                        OFL, WHICH ARE IGNORED FOR THIS TEST) BEING CLEAR (0).  IF THE
                                        CONTENTS OF TSSR ARE NOT AS EXPECTED, AN ERROR REPORT IS ISSUED
                                        LISTING THE EXPECTED DATA, ACTUAL DATA, AND THE DISCREPANCIES.
                                        THE ERROR REPORT ANALYZES THE TSSR CONTENTS AND DISCERNS AND
                                        REPORTS ONE OF THREE POSSIBILITIES:

                                             1.  TSSR CONTENTS ARE AMBIGUOUS (ANY OF BITS 11-14 ARE SET,
                                                 OR STATES OF SSR AND SC BITS DO NOT CORRESPOND TO THE
                                                 APPARENT ERROR CODE IN BITS 0-5):  INDICATES THAT THE
                                                 TSSR CONTENT CANNOT BE TRUSTED.  INDICATES A
                                                 CATASTROPHIC CONTROLLER MALFUNCTION.  THIS IS A FATAL
                                                 ERROR (EXECUTION IS ABORTED).  FIELD ACTION WOULD BE TO
                                                 REPLACE THE M7455.  IF THE M7455 ITSELF IS BEING
                                                 DEBUGGED, THE PROGRAM SHOULD BE RESTARTED WITH LOOP ON
                                                 ERROR ENABLED IN ORDER TO PROBE FOR THE PROBLEM.

                                             2.  SSR = 0, SC = 0 AND THE ERROR CODE IN BITS 0-5 IS IN
                                                 THE RANGE 17-13:  THIS IS A FATAL ERROR.  THE ERROR
                                                 CODE IS DECODED AND THE APPROPRIATE DESCRIPTION GIVEN.
                                                 INDICATES THAT A SERIOUS PROBLEM EXISTS.

023216                                  BGNTST
023216                                                          T1::
023216  012700  023414'                MOV     #TST1ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
023222  004737  016322'                JSR     PC,TSTSETUP     ;DO INITIAL TEST SETUP
023226  012737  000024  002214'        MOV     #20.,LOOPCNT    ;PERFORM 20 ITERATIONS
023234                          T1LOOP:
023234  005003                        CLR     R3              ;USE R3 AS FATAL ERROR FLAG

023236                                  BGNSUB                  ;////// BEGIN SUBTEST ////////
023236                                                          T1.1:
023236  104402                                                          TRAP    C$BSUB
```

```
 76
 77 023240                                   BRESET                                  ;ISSUE A BUS RESET
    023240  104433                                                                                        TRAP    C$RESET
 78 023242  004737  016060'                  JSR      PC,WAITF                       ;WAIT FOR READY
 79 023246  016501  000002                   MOV      TSSR(R5),R1                    ;GET THE CONTENTS OF TSSR
 80 023252  010102                           MOV      R1,R2                          ;CONTENTS OF TSSR
 81 023254  042702  176277                   BIC      #^C<HIADDR!OFL>,R2             ;THESE BITS MAY BE SET
 82 023260  052702  002200                   BIS      #SSR!NBA,R2                    ;READY AND NEW DATA SHOULD BE SET
 83 023264  020102                           CMP      R1,R2                          ;COMPARE EXPECTED TO RECEIVED
 84 023266  001405                           BEQ      10$                            ;BRANCH IF COMPARE
 88 023270                                   ERRDF    ERRNO,SFHERR,SFFMSG            ;REPORT A FATAL ERROR
    023270  104455                                                                                        TRAP    C$ERDF
    023272  000145                                                                                        .WORD   101
    023274  003677'                                                                                       .WORD   SFHERR
    023276  011712'                                                                                       .WORD   SFFMSG
 89 023300  005203                           INC      R3                             ;SET THE FATAL ERROR FLAG
 90 023302                          10$:
 91 023302                                   ENDSUB                                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    023302                                                                                           L10037:
    023302  104403                                                                                        TRAP    C$ESUB
 92
 93 023304  005703                           TST      R3                             ;DID WE HAVE FATAL ERROR ?
 94 023306  001402                           BEQ      20$                            ;BRANCH IF NOT
 95 023310  004737  017014'                  JSR      PC,CKDROP                      ;GO DROP THIS UNIT, IF ALLOWED
 96 023314  005003                  20$:     CLR      R3                             ;RESET FATAL ERROR FLAG
 97
 98
 99 023316                                   BGNSUB                                  ;///////////// BEGIN SUBTEST ////////////
    023316                                                                                           T1.2:
    023316  104402                                                                                        TRAP    C$BSUB
100
101 023320  005065  000002                   CLR      TSSR(R5)                       ;WRITE TO ISSUE A SOFT RESET
102 023324  004737  016060'                  JSR      PC,WAITF                       ;WAIT FOR READY TO SET
103 023330  016501  000002                   MOV      TSSR(R5),R1                    ;GET REGISTER TSSR DATA
104 023334  010102                           MOV      R1,R2                          ;CONTENTS OF TSSR
105 023336  042702  176277                   BIC      #^C<HIADDR!OFL>,R2             ;THESE BITS MAY BE SET
106 023342  052702  002200                   BIS      #SSR!NBA,R2                    ;READY AND NEW DATA SHOULD BE SET
107 023346  020102                           CMP      R1,R2                          ;COMPARE EXPECTED TO RECEIVED
108 023350  001405                           BEQ      10$                            ;BRANCH IF COMPARE
112 023352                                   ERRDF    ERRNO,SFIERR,SFFMSG            ;REPORT A FATAL ERROR
    023352  104455                                                                                        TRAP    C$ERDF
    023354  000146                                                                                        .WORD   102
    023356  003644'                                                                                       .WORD   SFIERR
    023360  011712'                                                                                       .WORD   SFFMSG
113 023362  005203                           INC      R3                             ;SET THE ERROR FLAG
114 023364                          10$:
115 023364                                   ENDSUB                                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    023364                                                                                           L10040:
    023364  104403                                                                                        TRAP    C$ESUB
116
117
118 023366  005703                           TST      R3                             ;FATAL ERROR DETECTED ?
119 023370  001402                           BEQ      20$                            ;BRANCH IF NOT
120 023372  004737  017014'                  JSR      PC,CKDROP                      ;SEE IF TIME TO DROP UNIT
121 023376  004737  016270'         20$:     JSR      PC,TSTLOOP                     ;SHOULD WE DO ITERATIONS ?
122 023402  103002                           BCC      40$                            ;BRANCH IF NOT
123 023404  000137  023234'                  JMP      T1LOOP                         ;LOOP UNTIL COUNT EXPIRED
```

```
231 023660  001405                        BEQ     35$                     ;ERROR IF NOT EQUAL
235 023662                                ERRHRD  ERRNO,T2TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
    023662  104456                                                            TRAP    C$ERHRD
    023664  000314                                                            .WORD   204
    023666  023720'                                                          .WORD   T2TSBA
    023670  015304'                                                          .WORD   EXPREC
236 023672  005203                        INC     R3                      ;FORCE SOFT INIT ON NEXT PASS
237
238 023674                          35$:   ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
    023674                                                                     10000$:
    023674  104405                                                            TRAP    C$ESEG
239
240 023676  105204                        INCB    R4                      ;NEXT DATA PATTERN TO TEST
241 023700  001270                        BNE     5$                      ;BRANCH TILL BACK TO ZERO
242 023702  004737  016270'               JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
243 023706  103002                        BCC     40$                     ;BRANCH IF NOT
244 023710  000137  023454'               JMP     T2LOOP                  ;LOOP UNTIL COUNT EXPIRED
245 023714                          40$:   EXIT    TST                     ;ALL DONE THIS TEST
    023714  104432                                                            TRAP    C$EXIT
    023716  000214                                                            .WORD   L10041-.
246
247                                 ;+
248                                 ;LOCAL TEXT MESSAGES FOR TEST
249                                 ;-
250
251 023720     124     123     102  T2TSBA: .ASCIZ  'TSBA Incorrect After TSDB High Write'
252 023765     124     123     123  T2TSSR: .ASCIZ  'TSSR Incorrect After TSDB High Write'
253 024032     116     157     040  T2SSR:  .ASCIZ  'No Sub-System Ready After TSDB High Write'
254 024104     127     162     141  TST2ID: .ASCIZ  'Wrap Data - High Byte'
255                                        .EVEN
256 024132                             ENDTST
    024132                                                                 L10041:
    024132  104401                                                            TRAP    C$ETST
257
258                                        .SBTTL  TEST  3: WRAP DATA - LOW BYTE
259
260                                 ;      THIS TEST FURTHER VERIFIES OPERATION OF MANY OF THE SAME
261                                 ;      ELEMENTS TESTED IN TEST 2, AND ADDITIONALLY VERIFIES:
262                                 ;
263                                 ;
264                                 ;           1.  LOW BYTE OF THE TSDB INPUT FILE REGISTER,
265                                 ;
266                                 ;           2.  LOW BYTE OF INTERNAL DAL BUS DRIVERS ON THE DC005
267                                 ;               TRANSCEIVER CIRCUITS,
268                                 ;
269                                 ;           3.  BASIC FUNCTIONING OF PARTS OF THE RAM,
270                                 ;
271                                 ;
272                                 ;
273                                 ;      THE PROGRAM WRITES A TEST DATA BYTE INTO THE LOW BYTE OF TSDB,
274                                 ;      WAITS FOR THE SSR BIT IN TSSR TO SET, THEN CHECKS THE CONTENTS
275                                 ;      OF BOTH TSBA AND TSSR.  THE MODULE IS FUNCTIONING CORRECTLY IF
276                                 ;      DATA WRITTEN APPEARS IN BOTH BYTES OF TSBA AND THE FINAL CONTENT
277                                 ;      OF TSSR IS CORRECT (SAME AS AFTER INITIALIZATION EXCEPT FOR BITS
278                                 ;      8 AND 9, WHICH SHOULD CONTAIN BITS 8 AND 9 OF THE DATA PATTERN
279                                 ;      WRITTEN.  AN ERROR IS REPORTED AND A DESCRIPTIVE ANALYSIS GIVEN
280                                 ;      IF A DISCREPANCY IN TSBA OR TSSR IS DETECTED.  THE ANALYSIS
```

```
281                                     ;       LISTS LIKELY FAULTY CANDIDATES FROM THE LOGIC ELEMENTS LISTED
282                                     ;       ABOVE.  THE TEST IS REPEATED FOR ALL COMBINATIONS OF TEST DATA
283                                     ;       BYTES (0-377 OCTAL).
284                                     ;
285 024134                                      BGNTST                                                  T3::
    024134
290 024134  012700  024577'                     MOV     #TST3ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
291 024140  004737  016322'                     JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
292 024144  012737  000024  002214'             MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
293 024152  005004                      T3LOOP: CLR     R4                      ;STARTING DATA PATTERN
294 024154  012703  000001                     MOV     #1,R3                   ;FORCE SOFT INIT ON FIRST PASS
295 024160  005703                      5$:     TST     R3                      ;SET IF INIT IS REQUIRED
296 024162  001412                             BEQ     10$                     ;BRANCH IF NO INIT NEEDED
297 024164  005003                             CLR     R3                      ;ASSUME NO INIT NEEDED ON NEXT LOOP
298 024166  004737  015604'                     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
299 024172  103406                             BCS     10$                     ;BR IF SOFT INIT = OK
303 024174  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
304 024176                                      ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
    024176  104455                                                                          TRAP    C$ERDF
    024200  000455                                                                         .WORD   301
    024202  003644'                                                                        .WORD   SFIERR
    024204  011644'                                                                        .WORD   SFIMSG
305 024206  005203                             INC     R3                      ;FORCE INIT ON NEXT PASS
306 024210  005037  002220'              10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
307
308 024214                                      BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
    024214  104404                                                                          TRAP    C$BSEG
309
310 024216  110465  000000                     MOVB    R4,TSDB(R5)             ;SET MAINT MODE + WRITE DATA
311 024222  004737  016060'                     JSR     PC,WAITF                ;WAIT FOR SSR TO SET
312 024226  103411                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
313 024230  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
314 024232  010402                             MOV     R4,R2                   ;DATA THAT WAS WRITTEN
318 024234                                      ERRDF   ERRNO,T3SSR,EXPREC      ;DEVICE FATAL SSR FAILED TO SET
    024234  104455                                                                          TRAP    C$ERDF
    024236  000456                                                                         .WORD   302
    024240  024526'                                                                        .WORD   T3SSR
    024242  015304'                                                                        .WORD   EXPREC
319 024244  005203                             INC     R3                      ;FORCE INIT ON NEXT PASS
320 024246  005237  002220'                     INC     FATFLG                  ;SET THE FATAL ERROR FLAG
321 024252                              15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    024252  104406                                                                          TRAP    C$CLP1
322 024254  005737  002220'                     TST     FATFLG                  ;DID WE GET FATAL ERROR ?
323 024260  001402                             BEQ     20$                     ;BRANCH IF NOT
324 024262  004737  017014'                     JSR     PC,CKDROP               ;DROP UNIT, IF DROP ALLOWED
325 024266  010402                      20$:    MOV     R4,R2                   ;DATA PATTERN WRITTEN
326 024270  042702  177774                     BIC     #^C<BIT0!BIT1>,R2       ;CLEAR ALL BUT LOW 2 BITS
327 024274  000302                             SWAB    R2                      ;BITS 8 AND 9 HAVE LOW DATA BITS
328 024276  052702  002200                     BIS     #SSR!NBA,R2             ;THESE BITS MUST BE SET ALSO
329 024302  016501  000002                     MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
330 024306  032701  000100                     BIT     #OFL,R1                 ;IS DRIVE OFF-LINE
331 024312  001402                             BEQ     25$                     ;BRANCH IF NOT OFFLINE
332 024314  052702  000100                     BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED
333 024320  020201                      25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
334 024322  001405                             BEQ     30$                     ;OKAY IF MATCH
338 024324                                      ERRHRD  ERRNO,T3TSSR,EXPREC     ;TSSR WASN'T CORRECT
    024324  104456                                                                          TRAP    C$ERHRD
```

```
                024326  000457                                                          .WORD   303
                024330  024462'                                                         .WORD   T3TSSR
                024332  015304'                                                         .WORD   EXPREC
    339 024334  005203                  INC     R3                  ;FORCE INIT ON NEXT PASS
    340 024336                  30$:    CKLOOP                      ;LOOP ON ERROR ?
                024336  104406                                                          TRAP    C$CLP1
    341 024340  016501  000000          MOV     TSBA(R5),R1         ;GET TSBA REGISTER CONTENTS
    342 024344  005002                  CLR     R2                  ;
    343 024346  150402                  BISB    R4,R2               ;DATA PATTERN WRITTEN
    344 024350  000302                  SWAB    R2                  ;MOVE INTO TOP BYTE
    345 024352  150402                  BISB    R4,R2               ;BOTH HALVES SHOULD BE SAME
    346 024354  020102                  CMP     R1,R2               ;COMPARE EXPECTED TO RECEIVED
    347 024356  001405                  BEQ     35$                 ;BR IF ERROR NOT EQUAL
    351 024360                          ERRHRD  ERRNO,T3TSBA,EXPREC ;PRINT THE ERROR & EXPD/RECV
                024360  104456                                                          TRAP    C$ERHRD
                024362  000460                                                          .WORD   304
                024364  024416'                                                         .WORD   T3TSBA
                024366  015304'                                                         .WORD   EXPREC
    352 024370  005203                  INC     R3                  ;FORCE INIT ON NEXT PASS
    353
    354 024372                  35$:    ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                024372                                                              10000$:
                024372  104405                                                          TRAP    C$ESEG
    355
    356 024374  105204                  INCB    R4                  ;NEXT DATA PATTERN TO TEST
    357 024376  001270                  BNE     5$                  ;ALL DONE WHEN BACK TO ZERO
    358 024400  004737  016270'         JSR     PC,TSTLOOP          ;SHOULD WE DO ITERATIONS ?
    359 024404  103002                  BCC     40$                 ;BRANCH IF NOT
    360 024406  000137  024152'         JMP     T3LOOP              ;LOOP UNTIL COUNT EXPIRED
    361 024412                  40$:    EXIT    TST                 ;ALL DONE THIS TEST
                024412  104432                                                          TRAP    C$EXIT
                024414  000210                                                          .WORD   L10042-.
    362
    363                             ;*
    364                             ;LOCAL TEXT MESSAGES FOR TEST
    365                             ;-
    366
    367 024416    124     123   102 T3TSBA: .ASCIZ  'TSBA Incorrect After TSDB Low Write'
    368 024462    124     123   123 T3TSSR: .ASCIZ  'TSSR Incorrect After TSDB Low Write'
    369 024526    116     157   040 T3SSR:  .ASCIZ  'No Sub-System Ready After TSDB Low Write'
    370 024577    127     162   141 TST3ID: .ASCIZ  'Wrap Data - Low Byte'
    371                             .EVEN
    372 024624                      ENDTST
                024624                                                          L10042:
                024624  104401                                                          TRAP    C$ETST
    373
    374                             .SBTTL  TEST  4: RAM TEST
    375
    376                             ;       THIS TEST VERIFIES THAT ALL LOCATIONS OF THE RAM ON THE M7455
    377                             ;       CAN PROPERLY STORE AND READ BACK ALL DATA PATTERNS, AND THAT
    378                             ;       EACH RAM LOCATION IS UNIQUELY ADDRESSED (I.E., THAT ONE AND ONLY
    379                             ;       ONE LOCATION IS ACCESSED BY ANY PARTICULAR ADDRESS).  THESE
    380                             ;       TESTS ARE PERFORMED BY THREE SUBTESTS, DESCRIBED BELOW.  A
    381                             ;       BYPRODUCT OF THESE TESTS IS A VERIFICATION OF TWO REGISTERS IN
    382                             ;       THE 2901 AND THE CAPABLITY OF THE 2901 TO CORRECTLY PERFORM AN
    383                             ;       ADD.
    384                             ;
```

```
                ;
                ;       TEST 4 , SUBTEST 1: -
                ;
                ;       THIS SUBTEST VERIFIES EACH RAM LOCATION BY FIRST PLACING THE
                ;       M7455 INTO MAINTENANCE MODE BY WRITING INTO THE LOW BYTE OF TSDB
                ;       AND THEN PERFORMING THE FOLLOWING SEQUENCE FOR EACH ADDRESS
                ;       0-7777 (OCTAL):
                ;
                ;            1.  THE ADDRESS TO BE TESTED IS LOADED INTO THE TSDB (VIA A
                ;                WORD WRITE).
                ;
                ;            2.  THE ADDRESSED RAM LOCATION IS WRITTEN, THEN READ INTO
                ;                THE LOW BYTE OF TSBA, BY WRITING A DATA BYTE INTO THE
                ;                LOW BYTE OF TSDB.
                ;
                ;            3.  THE LOW BYTE OF TSBA IS CHECKED TO SEE IF IT CONTAINS
                ;                THE DATA PATTERN ORIGINALLY WRITTEN;  A DISCREPANCY IS
                ;                REPORTED AS AN ERROR.
                ;
                ;            4.  THE ADDRESS OF THE LOCATION BEING TESTED IS AGAIN
                ;                WRITTEN INTO TSDB (WORD WRITE), TO CAUSE THE LOCATION
                ;                UNDER TEST TO AGAIN BE READ INTO THE LOW BYTE OF TSBA.
                ;                THE LOW BYTE OF TSBA IS AGAIN CHECKED AND DISCREPANCIES
                ;                REPORTED.
                ;
                ;            5.  THE HIGH BYTE OF TSBA IS CHECKED;  IT SHOULD CONTAIN
                ;                THE SUM OF THE HIGH AND LOW BYTES LAST WRITTEN INTO
                ;                TSDB AS A WORD.  A DISCREPANCY IS REPORTED AS A 2901
                ;                PROBLEM.
                ;
                ;            6.  THE CONTENT OF TSSR IS CHECKED;  SETTING OF THE SC BIT
                ;                IS IGNORED.  OTHER DISCREPANCIES IN TSSR ARE REPORTED.
                ;
                ;

                        BGNTST
                                                                        T4::
                        BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                        T4.1:
024626  104402                                                          TRAP    C$BSUB
024630  012700  026136'                MOV     #TST4ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
024634  004737  016322'                JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
024640  012737  000005  002214'        MOV     #5,LOOPCNT              ;PERFORM 5 ITERATIONS
024646                          T4LOOP:
024646  004737  015604'                JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
024652  103405                        BCS     20$                     ;BR IF INIT WAS OK
024654  010001                        MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
024656                                ERRDF   EPRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
024656  104455                                                          TRAP    C$ERDF
024660  000621                                                          .WORD   401
024662  003644'                                                         .WORD   SFIERR
024664  011644'                                                         .WORD   SFIMSG
024666  005004                  20$:   CLR     R4                      ;SET RAM ADDRESS AT ZERO
024670  004737  016146'                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
```

```
442 024674  105065  000000             CLRB    TSDB(R5)               ;SET INTO MAINTENANCE MODE
443 024700                      25$:
444 024700                             BGNSEG                         ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
    024700  104404                                                                                 TRAP    C$BSEG
445
446 024702  110402                     MOVB    R4,R2                  ;EXPECTED DATA FROM WRAP-AROUND
447 024704  004737  016146'            JSR     PC,CHKTSSR             ;WAIT FOR READY, NON-AMBIGUOUS
448 024710  010465  000000             MOV     R4,TSDB(R5)            ;LOAD ADDRESS INTO TSDB
449 024714  004737  016146'            JSR     PC,CHKTSSR             ;WAIT FOR READY, NON-AMBIGUOUS
450 024720  110265  000000             MOVB    R2,TSDB(R5)            ;LOADS DATA INTO RAM LOCATION
451 024724  004737  016146'            JSR     PC,CHKTSSR             ;WAIT FOR READY, NON-AMBIGUOUS
452 024730  016501  000000             MOV     TSBA(R5),R1            ;READS WRAP DATA
453 024734  120102                     CMPB    R1,R2                  ;DOES WRITTEN(WRAP) = READ
454 024736  001404                     BEQ     30$                    ;BR IF OK, THEY ARE EQUAL
458 024740                             ERRHRD  ERRNO,TSBAM2,EXPREC    ;DATA NOT WRAPPED CORRECTLY
    024740  104456                                                                                 TRAP    C$ERHRD
    024742  000622                                                                                 .WORD   402
    024744  025774'                                                                                .WORD   TSBAM2
    024746  015304'                                                                                .WORD   EXPREC
459 024750                      30$:
460 024750                             ENDSEG                         ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
    024750                                                                          10000$:
    024750  104405                                                                                 TRAP    C$ESEG
461
462 024752  005204                     INC     R4                     ;NEXT ADDRESS
463 024754  020427  010000             CMP     R4,#10000              ;END OF RAM MEMORY CHECK
464 024760  001347                     BNE     25$                    ;LOOP TILL ALL RAM WRITTEN
465 024762                             ESCAPE  SUB                    ;DON'T CONTINUE IF ERROR ON WRITE
    024762  104410                                                                                 TRAP    C$ESCAPE
    024764  000102                                                                                 .WORD   L10044-.
466
467 024766  005002                     CLR     R2                     ;CLEAR OUT R2 HIGH BITS
468 024770  005304                     DEC     R4                     ;SET BACK TO 7777
469 024772  110402                 40$: MOVB    R4,R2                  ;GET DATA PATTERN BACK IN SHAPE
470 024774  004737  016146'            JSR     PC,CHKTSSR             ;WAIT FOR READY, NON-AMBIGUOUS
471 025000  010465  000000             MOV     R4,TSDB(R5)            ;LOAD UP RAM ADDRESS POINTER
472 025004  004737  016146'            JSR     PC,CHKTSSR             ;WAIT FOR READY, NON-AMBIGUOUS
473 025010  016501  000000             MOV     TSBA(R5),R1            ;READ RAM CONTENTS BACK
474 025014  120102                     CMPB    R1,R2                  ;CHECK WITH DATA WRITTEN
475 025016  001404                     BEQ     45$                    ;BR IF OK, DATA IN = DATA OUT
479 025020                             ERRHRD  ERRNO,TSBAM2,EXPREC    ;WRITTEN DATA NOT = TO READ
    025020  104456                                                                                 TRAP    C$ERHRD
    025022  000623                                                                                 .WORD   403
    025024  025774'                                                                                .WORD   TSBAM2
    025026  015304'                                                                                .WORD   EXPREC
480 025030                      45$:   CKLOOP                         ;SCOPE LOOP
    025030  104406                                                                                 TRAP    C$CLP1
481 025032  116501  000001             MOVB    TSBAH(R5),R1           ;HIGH BYTE READ OF TSBA
482 025036  010402                     MOV     R4,R2                  ;DATA PATTERN WRITTEN
483 025040  000302                     SWAB    R2                     ;HIGH TO LOW
484 025042  060402                     ADD     R4,R2                  ;TOTAL OF BYTES IN LOW BYTE
485 025044  120102                     CMPB    R1,R2                  ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
486 025046  001404                     BEQ     50$                    ;BR IF OK, THEY SHOULD BE
490 025050                             ERRHRD  ERRNO,M2901,EXPREC     ;2901 PROBLEM ADDER
    025050  104456                                                                                 TRAP    C$ERHRD
    025052  000624                                                                                 .WORD   404
    025054  025704'                                                                                .WORD   M2901
```

```
                025056  015304'                                                 .WORD   EXPREC
491             025060                  50$:    CKLOOP                          ;SCOPE LOOP
                025060  104406                                                  TRAP    C$CLP1
492             025062  005304                  DEC     R4                      ;DROP DATA COUNTER (PATTERN)
493             025064  002342                  BGE     40$                     ;NOT AT LOC. ZERO YET
494
495             025066                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                025066                                                                  L10044:
                025066  104403                                                  TRAP    C$ESUB
496
497
498             025070                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
                025070                                                                  T4.2:
                025070  104402                                                  TRAP    C$BSUB
499                             ;               TEST 4, SUBTEST 2
500
501                             ;
502                             ;               THIS SUBTEST WRITES RAM WITH ALL ZEROS
503                             ;               THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
504                             ;
505             025072  004737  015604'          JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
506             025076  103405                  BCS     20$                     ;BR IF INIT WAS OK
510             025100  010001                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
511             025102                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
                025102  104455                                                  TRAP    C$ERDF
                025104  000625                                                  .WORD   405
                025106  003644'                                                 .WORD   SFIERR
                025110  011644'                                                 .WORD   SFIMSG
512             025112  005002          20$:    CLR     R2                      ;TEST DATA = 0
513             025114  005004                  CLR     R4                      ;STARTING RAM ADDRESS = 0
514             025116  004737  016146'        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
515             025122  105065  000000          CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
516             025126          25$:
517             025126                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
                025126  104404                                                  TRAP    C$BSEG
518
519             025130  004737  016146'        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
520             025134  010465  000000          MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
521             025140  004737  016146'        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
522             025144  110265  000000          MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
523             025150  004737  016146'        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
524             025154  016501  000000          MOV     TSBA(R5),R1             ;READS WRAP DATA
525             025160  120102                  CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
526             025162  001404                  BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
530             025164                          ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
                025164  104456                                                  TRAP    C$ERHRD
                025166  000626                                                  .WORD   406
                025170  025774'                                                 .WORD   TSBAM2
                025172  015304'                                                 .WORD   EXPREC
531             025174          30$:
532             025174                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                025174                                                                  10000$:
                025174  104405                                                  TRAP    C$ESEG
533
534             025176  005204                  INC     R4                      ;NEXT ADDRESS
535             025200  020427  010000          CMP     R4,#10000               ;END OF RAM MEMORY CHECK
536             025204  001350                  BNE     25$                     ;BR, MORE RAM TO GO
```

```
537
538 025206  005304                  35$:    DEC     R4                      ;SET BACK TO 7777
539 025210  005002                          CLR     R2                      ;SET TO ALL ZEROS
540 025212  004737  016146'          40$:    JSR     PC,CHKTSSR              ;WAIT FOR READY (SSR) TO SET
541 025216  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD UP THE ADDRESS FOR RAM
542 025222  004737  016146'                  JSR     PC,CHKTSSR              ;WAIT FOR READY (SSR) TO SET
543 025226  016501  000000                  MOV     TSBA(R5),R1             ;READ THE RAM CONTENTS BACK
544 025232  005002                          CLR     R2                      ;LOOKING FOR 000000 (EXPECTED)
545 025234  120102                          CMPB    R1,R2                   ;BOTH SHOULD BE 00000000 BINARY
546 025236  001404                          BEQ     43$                     ;BR, IF DATA IS GOOD
550 025240                                  ERRHRD  ERRNO,TSBAM3,EXPREC     ;CHARACTERISTICS DATA NOT CORRECT
    025240  104456                                                                      TRAP    C$ERHRD
    025242  000627                                                                      .WORD   407
    025244  026056'                                                                     .WORD   TSBAM3
    025246  015304'                                                                     .WORD   EXPREC
551 025250  012702  000377          43$:    MOV     #000377,R2              ;SET ALL ONES WORD
552 025254  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
553 025260  004737  016146'                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
554 025264  110265  000000                  MOVB    R2,TSDB(R5)             ;WRITE DATA INTO RAM
555 025270  004737  016146'                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
556 025274  016501  000000                  MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
557 025300  120102                          CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
558 025302  001404                          BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
562 025304                                  ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
    025304  104456                                                                      TRAP    C$ERHRD
    025306  000630                                                                      .WORD   408
    025310  025774'                                                                     .WORD   TSBAM2
    025312  015304'                                                                     .WORD   EXPREC
563 025314                          45$:    CKLOOP                          ;SCOPE LOOP
    025314  104406                                                                      TRAP    C$CLP1
564 025316  004737  016146'                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
565 025322  010465  000000                  MOV     R4,TSDB(R5)             ;WORD WRITE TO SET UP ADDRESS
566 025326  004737  016146'                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
567 025332  116501  000001                  MOVB    TSBAH(R5),R1            ;HIGH BYTE READ OF TSBA
568 025336  010403                          MOV     R4,R3                   ;DATA PATTERN WRITTEN
569 025340  000303                          SWAB    R3                      ;HIGH TO LOW
570 025342  060403                          ADD     R4,R3                   ;TOTAL OF BYTES IN LOW BYTE
571 025344  120103                          CMPB    R1,R3                   ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
572 025346  001404                          BEQ     50$                     ;BR IF OK, THEY SHOULD BE
576 025350                                  ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
    025350  104456                                                                      TRAP    C$ERHRD
    025352  000631                                                                      .WORD   409
    025354  025704'                                                                     .WORD   M2901
    025356  015304'                                                                     .WORD   EXPREC
577 025360                          50$:    CKLOOP                          ;SCOPE LOOP
    025360  104405                                                                      TRAP    C$CLP1
578 025362  005304                          DEC     R4                      ;DROP RAM ADDRESS POINTER
579 025364  002312                          BGE     40$                     ;NOT AT LOC. ZERO YET
580
581 025366                                  ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
    025366                                                                  L10045:
    025366  104403                                                                      TRAP    C$ESUB
582
583
584 025370                                  BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
    025370                                                                  T4.3:
    025370  104402                                                                      TRAP    C$BSUB
```

```
                                ;       TEST 4, SUBTEST 3
                                ;
                                ;       THIS SUBTEST WRITES RAM WITH ALL ONES
                                ;       THEN WALKS A ZERO WORD DOWN THROUGH MEMORY
                                ;
025372  004737  015604'          JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
025376  103405                  BCS     20$                     ;BR IF INIT WAS OK
025400  010001                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
025402                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
025402  104455                                                          TRAP    C$ERDF
025404  000632                                                          .WORD   410
025406  003644'                                                         .WORD   SFIERR
025410  011644'                                                         .WORD   SFIMSG
025412  012702  177777  20$:     MOV     #177777,R2              ;SET DATA AT ALL ONES
025416  005004                  CLR     R4                      ;SET RAM ADDRESS AT ZERO
025420  004737  016146'          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025424  105065  000000          CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
025430                  25$:
025430                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
025430  104404                                                          TRAP    C$BSEG
                                ;
025432  004737  016146'          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025436  010405  000000          MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
025442  004737  016146'          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025446  110265  000000          MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
025452  004737  016146'          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025456  016501  000000          MOV     TSBA(R5),R1             ;READS WRAP DATA
025462  120102                  CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
025464  001404                  BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
025466                          ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
025466  104456                                                          TRAP    C$ERHRD
025470  000633                                                          .WORD   411
025472  025774'                                                         .WORD   TSBAM2
025474  015304'                                                         .WORD   EXPREC
025476                  30$:
025476                          ENDSEG                          ;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
025476                                                                  10000$:
025476  104405                                                          TRAP    C$ESEG
                                ;
025500  005204                  INC     R4                      ;NEXT ADDRESS
025502  020427  010000          CMP     R4,#10000               ;END OF RAM MEMORY CHECK
025506  001350                  BNE     25$                     ;BR, MORE RAM TO GO
                                ;
025510                          ESCAPE  SUB                     ;NO CHECK IF WRITTEN INCORRECTLY
025510  104410                                                          TRAP    C$ESCAPE
025512  000152                                                          .WORD   L10046-.
025514  005304          35$:    DEC     R4                      ;SET BACK TO 7777
025516  004737  016146'  40$:    JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025522  012702  000377          MOV     #000377,R2              ;SET UP EXPECTED DATA REGISTER
025526  005001                  CLR     R1                      ;CLEAN OUT REGISTER
025530  010465  000000          MOV     R4,TSDB(R5)             ;SELECT ADDRESS IN RAM
025534  004737  016146'          JSR     PC,CHKTSSP              ;WAIT FOR READY (SSR)
025540  016501  000000          MOV     TSBA(R5),R1             ;PICK UP RAM CONTENTS
025544  120102                  CMPB    R1,R2                   ;IS MEMORY STILL ALL ONES
025546  001404                  BEQ     43$                     ;BR, IF OK (ALL ONES)
025550                          ERRHRD  ERRNO,TSBAM3,EXPREC     ;MEMORY CHANGED AFTER ALL ONES WRITE
```

```
TSVSA - HARDWARE TESTS  MACRO M1113  07-FEB-84 10:58                                                    SEQ 104
TEST  4: RAM TEST

        025550  104456                                                                  TRAP    C$ERHRD
        025552  000634                                                                  .WORD   412
        025554  026056'                                                                .WORD   TSBAM3
        025556  015304'                                                                .WORD   EXPREC
    638 025560  005002                  43$:    CLR     R2                      ;SET UP NEW EXPECTED
    639 025562  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
    640 025566  004737  016146'                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    641 025572  110265  000000                  MOVB    R2,TSDB(R5)             ;WRITE DATA INTO RAM
    642 025576  004737  016146'                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    643 025602  016501  000000                  MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
    644 025606  120102                          CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
    645 025610  001404                          BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
    649 025612                                  ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
        025612  104456                                                                  TRAP    C$ERHRD
        025614  000635                                                                  .WORD   413
        025616  025?14'                                                                .WORD   TSBAM2
        025620  015304'                                                                .WORD   EXPREC
    650 025622                  45$:    CKLOOP                                  ;SCOPE LOOP
        025622  104406                                                                  TRAP    C$CLP1
    651 025624  004737  016146'                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    652 025630  116501  000001                  MOVB    TSBAH(R5),R1            ;HIGH BYTE READ OF TSSA
    653 025634  010203                          MOV     R2,R3                   ;DATA PATTERN WRITTEN
    654 025636  000303                          SWAB    R3                      ;HIGH TO LOW
    655 025640  060203                          ADD     R2,R3                   ;TOTAL OF BYTES IN LOW BYTE
    656 025642  120103                          CMPB    R1,R3                   ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
    657 025644  001404                          BEQ     50$                     ;BR IF OK, THEY SHOULD BE
    661 025646                                  ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
        025646  104456                                                                  TRAP    C$ERHRD
        025650  000636                                                                  .WORD   414
        025652  025704'                                                                .WORD   M2901
        025654  015304'                                                                .WORD   EXPREC
    662 025656                  50$:    CKLOOP                                  ;SCOPE LOOP
        025656  104406                                                                  TRAP    C$CLP1
    663 025660  005304                          DEC     R4                      ;DROP RAM ADDRESS POINTER
    664 025662  001315                          BNE     40$                     ;NOT AT LOC. ZERO YET
    665
    666 025664                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        025664                                                                  L10046:
        025664  104403                                                                  TRAP    C$ESUB
    667
    668 025666  004737  016270'                JSR     PC,TSTLOOP              ;DO WE NEED TO ITERATE TEST ?
    669 025672  103002                          BCC     63$                     ;BRANCH IF NOT
    670 025674  000137  024646'                JMP     T4LOOP                  ;EXECUTE AGAIN
    671 025700                  63$:    EXIT    TST                             ;ALL DONE THIS TEST
        025700  104432                                                                  TRAP    C$EXIT
        025702  000256                                                                  .WORD   L10043-.
    672                                 ;+
    673                                 ;LOCAL TEXT MESSAGES FOR TEST
    674                                 ;-
    675
    676 025704      040     124     123 M2901:  .ASCIZ  ' TSBA High Byte Not Sum of Last TSDB Write (2901 Error)'
    677 025774      040     127     162 TSBAM2: .ASCIZ  ' Write to TSDB Not Equal to Read of TSBA Low Byte'
    678 026056      127     162     151 TSBAM3: .ASCIZ  'Write To RAM Location Modified Another Location'
    679 026136      122     101     115 TST4ID: .ASCIZ  'RAM Verification'
    680                                 .EVEN
    681 026160                          ENDTST
        026160                                                                  L10043:
```

```
                026160  104401                                                  TRAP    C$ETST
682
683                                             .SBTTL  TEST  5: SECOND INITIALIZATION TEST
684
685                                     ;       THIS TEST VERIFIES THE SAME ELEMENTS AS DID INITIALIZATION TEST
686                                     ;       #1 AND ALSO CHECKS THAT CERTAIN PARTS OF RAM IS CLEARED TO ZERO
687                                     ;       AND THAT 2901 REGISTERS 10 AND 11 ARE ALSO CLEARED TO ZERO.
688                                     ;       THIS IS A CONFIDENCE CHECK OF A PART OF THE SELF-TEST SEQUENCE
689                                     ;       (I.E., THAT IT IS REALLY BEING EXECUTED).  FOR EACH OF TWO
690                                     ;       SUBTESTS (ONE FOR INITIALIZING VIA A BUS INIT, THE OTHER FOR
691                                     ;       INITIALIZING BY WRITING INTO THE TSSR), THE FOLLOWING SEQUENCE
692                                     ;       IS PERFORMED:
693                                     ;
694                                     ;
695                                     ;            1.  EACH RAM LOCATION AND 2901 REGISTERS 10 AND 11 ARE SET
696                                     ;                TO ALL 1'S BY USING WRITES INTO THE TSDB REGISTER (LOW
697                                     ;                BYTE AND MAINTENANCE MODE WORD WRITES).
698                                     ;
699                                     ;            2.  THE CONTROLLER IS INITIALIZED AND THE VARIOUS CHECKS ON
700                                     ;                THE TSSR DESCRIBED IN INITIALIZATION TEST #1 ARE
701                                     ;                PERFORMED.
702                                     ;
703                                     ;            3.  #1'S (377 OCTAL) ARE WRITTEN INTO THE LOW BYTE OF TSDB,
704                                     ;                WHICH SHOULD CAUSE RAM LOCATION 0 TO BE WRITTEN TO ALL
705                                     ;                1'S SINCE 2901 REGISTERS 10 AND 11, SPECIFYING THE RAM
706                                     ;                ADDRESS, SHOULD BE 0.  RAM LOCATION 0 IS VERIFIED BY
707                                     ;                WRITING A WORD OF ZEROS INTO THE TSDB.  THE RESULTING
708                                     ;                LOW BYTE OF TSBA SHOULD CONTAIN ALL 1'S.
709                                     ;
710                                     ;            4.  THE ENTIRE RAM IS SCANNED.  LOCATION 0 SHOULD CONTAIN
711                                     ;                ALL 1'S AND THE REMAINING LOCATIONS, EXCEPT FOR THE
712                                     ;                MESSAGE BUFFER IMAGE AREA, SHOULD CONTAIN 0.
713                                     ;                DISCREPANCIES ARE REPORTED.  AN ERROR AT THIS POINT IS
714                                     ;                MOST LIKELY DUE TO A ROM, PIPELINE OR SEQUENCER PROBLEM
715                                     ;                OR A TIMING PROBLEM.
716                                     ;
717                                     ;
718                                     ;
719                                     ;
720 026162                                      BGNTST
    026162                                                                      T5::
725 026162  012700  027134'                     MOV     #TST5ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
726 026166  004737  016322'                     JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
727 026172  012737  000024  002214'             MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
728 026200                              TSLOOP:
729 026200  005037  002220'                     CLR     FATFLG                  ;CLEAR THE FATAL ERROR FLAG
730
731 026204                                      BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
    026204                                                                      T5.1:
    026204  104402                                                                      TRAP    C$BSUB
732
733 026206  004737  015604'                     JSR     PC,SOFINIT              ;DO A SOFT TO START
734 026212  103404                              BCS     10$                     ;BRANCH IF O.K.
738 026214                                      ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT ERROR AND DROP DRIVE
    026214  104455                                                                      TRAP    C$ERDF
    026216  000765                                                                      .WORD   501
    026220  003644'                                                                     .WORD   SFIERR
```

```
                026222  011644'                                                         .WORD   SFIMSG
    739 026224  012702  177777          10$:    MOV     #-1,R2          ;ALL ONE DATA PATTE
    740 026230  005004                  CLR     R4                      ;STARTING RAM ADDR
    741 026232  004737  016146'         JSR     PC,CHKTSSR              ;WAIT FOR READY, NO        UOUS
    742 026236  105065  000000          15$:    CLRB    TSDB(R5)        ;SET MAINTENANCE MO
    743 026242  004737  016146'         JSR     PC,CHKTSSR              ;WAIT FOR READY, NO        GUOUS
    744 026246  010465  000000          MOV     R4,TSDB(R5)             ;SET THE NEXT RAM ADDRESS
    745 026252  004737  016146'         JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    746 026256  110265  000000          MOVB    R2,TSDB(R5)             ;LOAD TEST DATA
    747 026262  005204                  INC     R4                      ;NEXT ADDRESS TO TEST
    748 026264  020427  007777          CMP     R4,#7777                ;COMPARE TO LAST ADDRESS
    749 026270  003762                  BLE     15$                     ;BRANCH TILL ALL DATA WRITTEN
    750 026272                          BRESET                          ;ISSUE A BUS RESET
                026272  104433                                                          TRAP    C$RESET
    751 026274  004737  016146'         JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    752 026300  016501  000002          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
    753 026304  010102                  MOV     R1,R2                   ;CONTENTS OF TSSR
    754 026306  042702  176277          BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
    755 026312  052702  002200          BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
    756 026316  020102                  CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
    757 026320  001406                  BEQ     20$                     ;BRANCH IF COMPARE
    761 026322                          ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
                026322  104455                                                          TRAP    C$ERDF
                026324  000766                                                          .WORD   502
                026326  003677'                                                        .WORD   SFHERR
                026330  011712'                                                        .WORD   SFFMSG
    762 026332  005237  002220'         INC     FATFLG                  ;SET FATAL ERROR FLAG
    763 026336                  20$:    CKLOOP                          ;LOOP ON ERROR IF FLAG SET
                026336  104406                                                          TRAP    C$CLP1
    764 026340                          ESCAPE  SUB                     ;EXIT IF FATAL ERROR DETECTED
                026340  104410                                                          TRAP    C$ESCAPE
                026342  000170                                                          .WORD   L10050-.
    765 026344  004737  016146'         JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
    766 026350  105065  000000          CLRB    TSDB(R5)                ;PUT BACK INTO MAINTENANCE MODE
    767 026354  004737  016146'         JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    768 026360  005065  000000          CLR     TSDB(R5)                ;SET ADDRESS BACK TO 0000
    769 026364  012702  000377          MOV     #377,R2                 ;
    770 026370  004737  016146'         JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    771 026374  110265  000000          MOVB    R2,TSDB(R5)             ;SHOULD POINT TO RAM 0
    772 026400  004737  016146'         JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    773 026404  005065  000000          CLR     TSDB(R5)                ;SELECT LOCATION 0
    774 026410  004737  016146'         JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    775 026414  116501  000000          MOVB    TSBA(R5),R1             ;READ RAM LOCATION SPECIFIED
    776 026420  120102                  CMPB    R1,R2                   ;LOCATION SHOULD BE 377 OCTAL
    777 026422  001406                  BEQ     25$                     ;BR IF OK
    778 026424                          ERRDF   ERRNO,T5ADDR,EXPREC     ;WASN'T POINTING TO CORRECT LOC.
                026424  104455                                                          TRAP    C$ERDF
                026426  000766                                                          .WORD   502
                026430  027222'                                                        .WORD   T5ADDR
                026432  015304'                                                        .WORD   EXPREC
    779 026434  005237  002220'         INC     FATFLG                  ;SET THE FATAL ERROR FLAG
    780 026440                  25$:    CKLOOP                          ;SCOPE LOOP
                026440  104406                                                          TRAP    C$CLP1
    781 026442                          ESCAPE  SUB                     ;NO MORE CHECKS IF FATAL ERROR
                026442  104410                                                          TRAP    C$ESCAPE
                026444  000066                                                          .WORD   L10050-.
    782 026446  012704  000310          MOV     #310,R4                 ;START WITH LOC 310
```

```
                027066  104410                                                          TRAP    C$ESCAPE
                027070  000012                                                          .WORD   L10051-.
    869         027072  005204                  INC     R4              ;LOOK AT NEXT RAM LOC.
    870         027074  020427  000400          CMP     R4,#400         ;AT TOP OF RAM ADDRESS SPACE
    871         027100  001353                  BNE     30$             ;BRANCH TILL ALL MEMORY TESTED
    872
    873         027102                          ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                027102                                                                  L10051:
                027102  104403                                                          TRAP    C$ESUB
    874
    875         027104  005737  002220'          TST     FATFLG          ;IS FATAL ERROR FLAG SET ?
    876         027110  001402                  BEQ     50$             ;BRANCH IF NOT
    877         027112  004737  017014'          JSR     PC,CKDROP       ;NO LOOP, TRY TO DROP DEVICE
    878         027116  004737  016270'  50$:    JSR     PC,TSTLOOP      ;SHOULD WE DO ITERATIONS ?
    879         027122  103002                  BCC     60$             ;BRANCH IF NOT
    880         027124  000137  026200'          JMP     TSLOOP          ;LOOP UNTIL COUNT EXPIRED
    881         027130          60$:    EXIT    TST                     ;ALL DONE THIS TEST
                027130  104432                                                          TRAP    C$EXIT
                027132  000132                                                          .WORD   L10047-.
    882
    883                                 ;*
    884                                 ;LOCAL TEXT MESSAGES FOR TEST
    885                                 ;-
    886
    887 027134      105     170     164 TST5ID: .ASCIZ  'Extended Initialization'
    888 027164      111     156     143 T5MEM:  .ASCIZ  'Incorrect RAM Data After Init'
    889 027222      111     156     143 T5ADDR: .ASCIZ  'Incorrect RAM Address After Init'
    890                                 .EVEN
    891 027264                          ENDTST
                027264                                                          L10047:
                027264  104401                                                          TRAP    C$ETST
    892
    893
    894                                 .SBTTL  TEST  6: COMMAND REJECT
    895
    896
    897                         ;       THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE
    898                         ;       CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS
    899                         ;       (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA AND TSSR
    900                         ;       REGISTERS ARE LEFT IN THE PROPER STATE AFTER EACH COMMAND IS
    901                         ;       REJECTED.  THIS TEST CHECKS MICROPROCESSOR SEQUENCING, BASIC
    902                         ;       COMMAND DECODING AND DATI DMA HANDLING.  THIS TEST CONTAINS TWO
    903                         ;       SUBTESTS:  SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER
    904                         ;       THAN WRITE CHARACTERISTICS) WITH THE INTERRUPT ENABLE (IE) BIT
    905                         ;       CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE
    906                         ;       REJECTED COMMAND;  SUBTEST 2 PERFORMS SIMILARLY TO SUBTEST 1 BUT
    907                         ;       SETS THE IE BIT IN EACH COMMAND WORD AND VERIFIES THAT AN
    908                         ;       INTERRUPT IS GENERATED WHEN THE COMMAND IS REJECTED.  SUBTEST 1
    909                         ;       SETS UP THE INTERRUPT SERVICE ROUTINE TO FLAG UNEXPECTED
    910                         ;       INTERRUPTS.  THE COMMAND WORD IN THE COMMAND BUFFER IS
    911                         ;       INITIALIZED TO 100000 (OCTAL) AND THE REMAINING THREE WORDS IN
    912                         ;       THE COMMAND BUFFER ARE SET TO KNOWN UNIQUE PATTERNS.  THEN THE
    913                         ;       FOLLOWING SEQUENCE IS PERFORMED:
    914                         ;
    915                         ;
    916                         ;           1.  INITIALIZE THE CONTROLLER BY WRITING INTO THE TSSR;
    917                         ;               PROPER INITIAL CONDITIONS ARE VERIFIED.
```

```
                                        ;
                                        ;          2.  TSDB IS WRITTEN WITH ADDRESS OF THE COMMAND BUFFER TO
                                        ;              START PROCESSING.
                                        ;
                                        ;          3.  THE PROGRAM WAITS FOR SSR TO SET;  IF SSR DOES NOT SET,
                                        ;              AN ERROR REPORT IS ISSUED AND THE TEST IS ABORTED.
                                        ;
                                        ;          4.  THE CONTENTS OF TSSR ARE CHECKED.  TSSR IS CORRECT IF
                                        ;              IT CONTAINS EITHER OCTAL 102206 OR 102306 (BIT 6
                                        ;              DEPENDS UPON THE STATE OF THE TAPE TRANSPORT).
                                        ;
                                        ;          5.  THE CONTENTS OF TSBA ARE CHECKED.  TSBA SHOULD CONTAIN
                                        ;              THE INITIAL COMMAND BUFFER ADDRESS (LOADED IN STEP 2)
                                        ;              PLUS 10 (OCTAL);  I.E., TSBA SHOULD POINT TO THE WORD
                                        ;              JUST AFTER THE COMMAND PACKET (NOTE THAT 4 COMMAND
                                        ;              PACKET WORDS ARE ALWAYS FETCHED).
                                        ;
                                        ;          6.  USING THE MAINTENANCE MODE WRAPAROUND FUNCTIONS, THE
                                        ;              COMMAND IMAGE BLOCK IN THE M7455'S RAM (LOCATIONS
                                        ;              201-210 (OCTAL)) ARE CHECKED;  THE IMAGE SHOULD CONTAIN
                                        ;              A COPY OF THE FOUR COMMAND PACKET WORDS AS SET UP IN
                                        ;              CPU MEMORY.
                                        ;
                                        ;          7.  THE COMMAND WORD IN THE COMMAND BUFFER IS INCREMENTED
                                        ;              TO THE NEXT PATTERN NOT CONTAINING WRITE
                                        ;              CHARACTERISTICS OR IE.  THE REMAINING THREE WORD OF THE
                                        ;              COMMAND BUFFER ARE SEQUENCED WITH PSEUDO-RANDOM DATA.
                                        ;              IF THE COMMAND WORD HAS NOT REACHED ITS MAXIMUM VALUE
                                        ;              (177777+1), THE TEST SEQUENCE IS REPEATED.
                                        ;
                                        ;
                                        ;
                                        ;      SUBTEST 2 IS IDENTICAL TO SUBTEST 1, EXCEPT THAT THE PROGRAM
                                        ;      CAUSES THE IE BIT TO BE SET IN EACH COMMAND WORD AND THEN
                                        ;      VERIFIES THAT AN INTERRUPT OCCURS.
                                        ;
027266                                          BGNTST
027266                                                                          T6::
027266  012700  030523'                         MOV     #TST6ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
027272  004737  016322'                         JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
027276  012737  000024  002214'                 MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
027304                                  T6LOOP:
027304                                          BGNSUB                          ;///////////// BEGIN SUBTEST ////////////
027304                                                                          T6.1:
027304  104402                                                                          TRAP    C$BSUB

027306                                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
027306  012700  000000                                                                  MOV     #PRI00,R0
027312  104441                                                                          TRAP    C$SPRI
027314  012704  030200'                         MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
027320  012703  002750'                         MOV     #TSTBLK,R3              ;BLOCK OF TEST DATA
027324  012314                          5$:     MOV     (R3)+,(R4)              ;INSERT THE NEXT TEST DATA WORD
027326                                          BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
027326  104404                                                                          TRAP    C$BSEG
027330  004737  015604'                         JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
027334  103405                                  BCS     10$                     ;BR IF SOFT INIT = OK
027336  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
  976 027340                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      027340  104455                                                                TRAP    C$ERDF
      027342  001131                                                                .WORD   601
      027344  003644'                                                              .WORD   SFIERR
      027346  011644'                                                              .WORD   SFIMSG
  977 027350  005037  002220'           10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
  978 027354  005037  002222'                   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
  979 027360  004737  016146'                   JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
  980 027364  042714  000200                    BIC     #BIT7,(R4)              ;DISABLE INTERRUPTS
  981 027370  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
  982 027374  004737  016060'                   JSR     PC,WAITF                ;WAIT FOR SSR TO SET
  983 027400  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
  984 027402  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  988 027404                                    ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      027404  104455                                                                TRAP    C$ERDF
      027406  001132                                                                .WORD   602
      027410  030235'                                                              .WORD   T6SSR
      027412  011656'                                                              .WORD   PKTSSR
  989 027414  005237  002220'                   INC     FATFLG                  ;SET FATAL ERROR FLAG
  990 027420                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      027420  104406                                                                TRAP    C$CLP1
  991 027422                                    ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
      027422  104410                                                                TRAP    C$ESCAPE
      027424  000170                                                                .WORD   L10053-.
  992 027426  005737  002222'                   TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
  993 027432  001404                            BEQ     22$                     ;BRANCH IF NOT
  997 027434                                    ERRHRD  ERRNO,T6INT,PKTSSR
      027434  104456                                                                TRAP    C$ERHRD
      027436  001133                                                                .WORD   603
      027440  030313'                                                              .WORD   T6INT
      027442  011656'                                                              .WORD   PKTSSR
  998 027444  012702  102206            22$:    MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
  999 027450  004737  016146'                   JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
 1000 027454  016501  000002                    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 1001 027460  032701  000100                    BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 1002 027464  001402                            BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 1003 027466  052702  000100                    BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 1004 027472  020201                    25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 1005 027474  001404                            BEQ     30$                     ;OKAY IF MATCH
 1009 027476                                    ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT SET TO REJECT
      027476  104456                                                                TRAP    C$ERHRD
      027500  001134                                                                .WORD   604
      027502  030210'                                                              .WORD   T6NBA
      027504  011656'                                                              .WORD   PKTSSR
 1010 027506                            30$:    CKLOOP                          ;LOOP ON ERROR ?
      027506  104406                                                                TRAP    C$CLP1
 1011 027510  004737  016146'                   JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
 1012 027514  016501  000000                    MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
 1013 027520  010402                            MOV     R4,R2                   ;START OF THE PACKET
 1014 027522  062702  000010                    ADD     #10,R2                  ;EXPECT TSDA TO PACKET + 10
 1015 027526  020102                            CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
 1016 027530  001404                            BEQ     35$                     ;ERROR IF NOT EQUAL
 1020 027532                                    ERRHRD  ERRNO,T6TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
      027532  104456                                                                TRAP    C$ERHRD
      027534  001135                                                                .WORD   605
      027536  030451'                                                              .WORD   T6TSBA
      027540  015304'                                                              .WORD   EXPREC
```

```
1021
1022
1023 027542  004737  010724'        35$:   JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
1024 027546  103404                        BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
1028 027550                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     027550  104456                                                               TRAP    C$ERHRD
     027552  001136                                                             .WORD   606
     027554  004737'                                                           .WORD   PKTRAM
     027556  015320'                                                           .WORD   RAMERR
1029 027560                        40$:    ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<
     027560                                                                     10000$:
     027560  104405                                                             TRAP    C$ESEG
1030 027562  011300                        MOV     (R3),R0                 ;PACKET COMMAND WORD
1031 027564  042700  177740              BIC     #177740,R0              ;GET BITS 0-4
1032 027570  020027  000004              CMP     R0,#4                   ;DON'T TEST WRITE CHARACTERISTICS
1033 027574  001002                        BNE     45$                     ;BRANCH IF OK
1034 027576  062703  000002              ADD     #2,R3                   ;GET NEXT WORD FROM DATA TABLE
1035 027602  020327  003060'        45$:   CMP     R3,#TBLEND              ;REACHED END OF TABLE ?
1036 027606  103002                        BHIS    50$                     ;BRANCH IF END OF TABLE
1037 027610  000137  027324'              JMP     5$                      ;CONTINUE TEST WITH NEW DATA
1038
1039 027614                        50$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     027614                                                                     L10053:
     027614  104403                                                             TRAP    C$ESUB
1040
1041 027616  005737  002220'              TST     FATFLG                  ;ANY FATAL ERRORS ?
1042 027622  001402                        BEQ     60$                     ;BRANCH IF NOT
1043 027624  004737  017014'              JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
1044
1045 027630                        60$:    BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     027630                                                                     T6.2:
     027630  104402                                                             TRAP    C$BSUB
1046
1047 027632                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     027632  012700  000000                                                     MOV     #PRI00,R0
     027636  104441                                                             TRAP    C$SPRI
1048 027640  012704  030200              MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1049 027644  012703  002750'              MOV     #TSTBLK,R3              ;START OF TEST DATA
1050 027650  012314                  5$:   MOV     (R3)+,(R4)              ;PLACE NEXT DATA WORD IN PACKET
1051 027652                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     027652  104404                                                             TRAP    C$BSEG
1052 027654  004737  015604'              JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1053 027660  103405                        BCS     10$                     ;BR IF SOFT INIT = OK
1057 027662  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1058 027664                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     027664  104455                                                             TRAP    C$ERDF
     027666  001137                                                           .WORD   607
     027670  003644'                                                         .WORD   SFIERR
     027672  011644'                                                         .WORD   SFIMSG
1059 027674  005037  002220'        10$:   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1060 027700  005037  002222'              CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1061 027704  004737  016146'              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1062 027710  052714  000200              BIS     #BIT7,(R4)              ;ENABLE INTERRUPTS
1063 027714  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1064 027720  004737  016060'              JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1065 027724  103407                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1066 027726  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
1070 027730                                    ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     027730  104455                                                                TRAP    C$ERDF
     027732  001140                                                                .WORD   608
     027734  030235'                                                              .WORD   T6SSR
     027736  011656'                                                              .WORD   PKTSSR
1071 027740  005237  002220'                   INC     FATFLG                  ;SET FATAL ERROR FLAG
1072 027744                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027744  104406                                                                TRAP    C$CLP1
1073 027746                                    ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     027746  104410                                                                TRAP    C$ESCAPE
     027750  000170                                                                .WORD   L10054-.
1074 027752  005737  002222'                   TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1075 027756  001004                            BNE     22$                     ;BRANCH IF YES
1079 027760                                    ERRHRD  ERRNO,T6NINT,PKTSSR     ;REPORT ERROR IF NO INTERRUPT
     027760  104456                                                                TRAP    C$ERHRD
     027762  001141                                                                .WORD   609
     027764  030371'                                                              .WORD   T6NINT
     027766  011656'                                                              .WORD   PKTSSR
1080 027770  012702  102206               22$:    MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
1081 027774  004737  016146'                   JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1082 030000  016501  000002                   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1083 030004  032701  000100                   BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1084 030010  001402                            BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1085 030012  052702  000100                   BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1086 030016  020201                    25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1087 030020  001404                            BEQ     30$                     ;OKAY IF MATCH
1091 030022                                    ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT SET TO REJECT
     030022  104456                                                                TRAP    C$ERHRD
     030024  001142                                                                .WORD   610
     030026  030210'                                                              .WORD   T6NBA
     030030  011656'                                                              .WORD   PKTSSR
1092 030032                            30$:    CKLOOP                          ;LOOP ON ERROR ?
     030032  104406                                                                TRAP    C$CLP1
1093 030034  004737  016146'                   JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1094 030040  016501  000000                   MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1095 030044  010402                            MOV     R4,R2                   ;START OF THE PACKET
1096 030046  062702  000010                   ADD     #10,R2                  ;EXPECT TSDA TO PACKET + 10
1097 030052  020102                            CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1098 030054  001404                            BEQ     35$                     ;ERROR IF NOT EQUAL
1102 030056                                    ERRHRD  ERRNO,T6TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     030056  104456                                                                TRAP    C$ERHRD
     030060  001143                                                                .WORD   611
     030062  030451'                                                              .WORD   T6TSBA
     030064  015304'                                                              .WORD   EXPREC
1103
1104
1105 030066  004737  010724'               35$:    JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
1106 030072  103404                            BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
1110 030074                                    ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     030074  104456                                                                TRAP    C$ERHRD
     030076  001144                                                                .WORD   612
     030100  004737'                                                              .WORD   PKTRAM
     030102  015320'                                                              .WORD   RAMERR
1111 030104                            40$:    ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     030104                                                                                 10000$:
     030104  104405                                                                TRAP    C$ESEG
1112 030106  011300                            MOV     (R3),R0                 ;NEXT PACKET CONTAINED WORD
```

```
1113 030110  042700  177740              BIC     #177740,R0          ;GET BITS 0-4
1114 030114  020027  000004              CMP     R0,#4               ;DON'T TEST WRITE CHARACTERISTICS
1115 030120  001002                      BNE     45$                 ;BRANCH IF NOT WRITE CHARACTERISTICS
1116 030122  062703  000002              ADD     #2,R3               ;BY-PASS WRITE CHARACTERISTCS
1117 030126  020327  003060'    45$:     CMP     R3,#TBLEND          ;HAVE WE COMPLETED DATA TABLE ?
1118 030132  103002                      BHIS    50$                 ;BRANCH IF ALL TESTED
1119 030134  000137  027650'            JMP     5$                  ;TEST WITH NEXT DATA
1120
1121 030140              50$:    ENDSUB                              ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     030140                                                                  L10054:
     030140  104403                                                          TRAP    C$ESUB
1122 030142  005737  002220'            TST     FATFLG              ;ANY FATAL ERRORS ?
1123 030146  001402                      BEQ     60$                 ;BRANCH IF NOT
1124 030150  004737  017014'            JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
1125 030154  004737  016270'    60$:    JSR     PC,TSTLOOP          ;SHOULD WE DO ITERATIONS ?
1126 030160  103002                      BCC     62$                 ;BRANCH IF NOT
1127 030162  000137  027304'            JMP     T6LOOP              ;LOOP UNTIL COUNT EXPIRED
1128 030166              62$:    EXIT    TST                         ;ALL DONE THIS TEST
     030166  104432                                                  TRAP    C$EXIT
     030170  000352                                                  .WORD   L10052-.
1129
1130                             ;+
1131                             ;LOCAL STORAGE FOR THIS TEST
1132                             ;-
1133
1135 030172                              .BLKB   10-<.-TSV2&7>
1137 030200              T6PACKET:                                   ;COMMAND PACKET FOR TEST
1138 030200  000000                      .WORD   0                   ;WILL CONTAIN VARIABLE COMMANDS
1139 030202  052525                      .WORD   052525
1140 030204  125252                      .WORD   125252
1141 030206  052525                      .WORD   052525
1142
1143
1144                             ;+
1145                             ;LOCAL TEXT MESSAGES FOR TEST
1146                             ;-
1147
1148 030210     103     157     155  T6NBA:  .ASCIZ  'Command Not Rejected'
1149 030235     103     157     156  T6SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Packet'
1150 030313     125     156     145  T6INT:  .ASCIZ  'Unexpected Interrupt Received On Write Packet'
1151 030371     105     170     160  T6NINT: .ASCIZ  'Expected Interrupt Not Received On Write Packet'
1152 030451     111     156     143  T6TSBA: .ASCIZ  'Incorrect TSBA Address After Packet Write'
1153 030523     103     157     155  TST6ID: .ASCIZ  'Command Reject'
1154                                     .EVEN
1155 030542             ENDTST
     030542                                                          L10052:
     030542  104401                                                          TRAP    C$ETST
1156
1157                                     .SBTTL  TEST  7: WRITE CHARACTERISTICS
1158
1159
1160                             ;       THIS TEST VERIFIES BASIC OPERATION OF THE WRITE CHARACTERISTICS
1161                             ;       COMMAND.  IT VERIFIES THAT THE COMMAND BLOCK AND CHARACTERISTICS
1162                             ;       DATA BLOCK ARE FETCHED PROPERLY FROM CPU MEMORY, THE NEED BUFFER
1163                             ;       ADDRESS (NBA) BIT IN TSSR IS HANDLED PROPERLY, AND THAT A PROPER
1164                             ;       MESSAGE PACKET IS STORED, WHERE APPROPRIATE.  THIS TEST DOES NOT
1165                             ;       CHECK THAT THE VARIOUS FUNCTIONS ENABLED BY CHARACTERISTIC MODE
```

```
1166                                    ;       DATA BITS OPERATE PROPERLY;  THE FUNCTIONING OF THESE BITS IS
1167                                    ;       VERIFIED IN SUBSEQUENT TESTS.  ALL COMMANDS EXECUTED IN THIS
1168                                    ;       TEST HAVE THE INTERRUPT ENABLE (IE) BIT CLEARED TO ZERO, SO NO
1169                                    ;       INTERRUPTS SHOULD BE GENERATED.  HOWEVER, THE PROGRAM RUNS AT
1170                                    ;       PROCESSOR PRIORITY 0, WITH THE INTERRUPT SERVICE ROUTINE SET UP
1171                                    ;       TO FLAG UNEXPECTED INTERRUPTS.  IF AN INTERRUPT OCCURS, A
1172                                    ;       PROBLEM EXISTS IN EITHER THE PDP-11 BUS INTERFACE SECTION OR IN
1173                                    ;       THE ROM OR PIPELINE.
1174                                    ;
1175                                    ;       THIS TEST CHECKS VARIOUS MICROPROGRAM SEQUENCES, COMMAND
1176                                    ;       DECODING, DMA LOGIC, AND BASIC PACKET PROTOCOL HANDLING.  THIS
1177                                    ;       IS THE FIRST TEST IN WHICH DATO DMA CYCLES (FOR STORING THE
1178                                    ;       MESSAGE PACKET) ARE PERFORMED.  ANY ERRORS IN THE BODY OF THE
1179                                    ;       TEST (I.E, ERRORS OTHER THAN INITIALIZATION ERRORS RELATED TO
1180                                    ;       THE TRANSPORT BUS) DEFINITELY INDICATE A BAD M7455 MODULE.
1181                                    ;
1182                                    ;
1183                                    ;
1184 030544                                    BGNTST
     030544                                                                                  T7::
1189 030544  012700  034033'                   MOV     #TST7ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
1190 030550  004737  016322'                   JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
1191 030554  012737  000024  002214'          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
1192 030562                             T7LOOP:
1193 030562                                    BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
     030562                                                                                  T7.1:
     030562  104402                                                                          TRAP    C$BSUB
1194 030564  004737  034062'                   JSR     PC,T7REST               ;SET PACKET TO START-UP VALUES
1195
1196 030570                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     030570  012700  000000                                                                  MOV     #PRI00,R0
     030574  104441                                                                          TRAP    C$SPRI
1197 030576  012703  002762'                   MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
1198 030602  012704  032700'                   MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1199 030606  012764  000010  000006           MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
1200 030614                             5$:
1201 030614                                    BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     030614  104404                                                                          TRAP    C$BSEG
1202
1203 030616  004737  015604'                   JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1204 030622  103405                           BCS     10$                     ;BR IF SOFT INIT = OK
1208 030624  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1209 030626                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     030626  104455                                                                          TRAP    C$ERDF
     030630  001275                                                                          .WORD   701
     030632  003644'                                                                         .WORD   SFIERR
     030634  011644'                                                                         .WORD   SFIMSG
1210 030636  005037  002220'             10$:  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1211 030642  005037  002222'                   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1212 030646  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1213 030652  004737  016146'                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1214 030656  103407                           BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1215 030660  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1219 030662                                    ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     030662  104455                                                                          TRAP    C$ERDF
     030664  001276                                                                          .WORD   702
     030666  033501'                                                                         .WORD   T7SSR
```

```
                030670  011656'                                                                 .WORD   PKTSSR
1220            030672  005237  002220'         INC     FATFLG          ;SET FATAL ERROR FLAG
1221            030676                  15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                030676  104406                                                                  TRAP    C$CLP1
1222            030700                          ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
                030700  104410                                                                  TRAP    C$ESCAPE
                030702  000152                                                                  .WORD   10000$-.
1223            030704  005737  002222'         TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
1224            030710  001404                  BEQ     22$             ;BRANCH IF NOT
1228            030712                          ERRHRD  ERRNO,T7INT,PKTSSR
                030712  104456                                                                  TRAP    C$ERHRD
                030714  001277                                                                  .WORD   703
                030716  033661'                                                                 .WORD   T7INT
                030720  011656'                                                                 .WORD   PKTSSR
1229            030722  016501  000002          22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
1230            030726  012702  000200                  MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
1231            030732  032701  000100                  BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
1232            030736  001402                  BEQ     25$             ;BRANCH IF NOT OFF-LINE
1233            030740  052702  000100                  BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
1234            030744  020201                  25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
1235            030746  001404                  BEQ     30$             ;OKAY IF MATCH
1239            030750                          ERRHRD  ERRNO,T7NBA,PKTSSR      ;NBA NOT ZERO
                030750  104456                                                                  TRAP    C$ERHRD
                030752  001300                                                                  .WORD   704
                030754  033040'                                                                 .WORD   T7NBA
                030756  011656'                                                                 .WORD   PKTSSR
1240            030760                  30$:    CKLOOP                  ;LOOP ON ERROR ?
                030760  104406                                                                  TRAP    C$CLP1
1241            030762  004737  016146'         JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
1242            030766  016501  000000          MOV     TSBA(R5),R1     ;GET TSBA REGISTER CONTENTS
1243            030772  012702  032726'         MOV     #T7BFR,R2       ;START OF THE DATA BUFFER
1244            030776  032762  000200  000012  BIT     #BIT7,XST2(R2)  ;IS EXTENDED FEATURES BIT SET ?
1245            031004  001404                  BEQ     32$             ;BRANCH IF EXTENDED FEATURES OFF
1246            031006  005237  002224'         INC     EXTFEA          ;SET EXTENDED FEATURES FOR SUBTEST 6
1247            031012  062702  000002          ADD     #2,R2           ;EXTRA WORD IF SPECIAL FEATURES
1248            031016  062702  000016  32$:    ADD     #14.,R2         ;EXPECTED CONTENTS OF TSDA
1249            031022  020102                  CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
1250            031024  001404                  BEQ     35$             ;ERROR IF NOT EQUAL
1254            031026                          ERRHRD  ERRNO,T7TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
                031026  104456                                                                  TRAP    C$ERHRD
                031030  001301                                                                  .WORD   705
                031032  033750'                                                                 .WORD   T7TSBA
                031034  015304'                                                                 .WORD   EXPREC
1255
1256
1257            031036  004737  010724'         35$:    JSR     PC,CKRAM        ;SEE IF DATA IN RAM IS CORRECT
1258            031042  103404                  BCS     40$             ;BRANCH IF PACKET IN RAM IS CORRECT
1262            031044                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
                031044  104456                                                                  TRAP    C$ERHRD
                031046  001302                                                                  .WORD   706
                031050  004737'                                                                 .WORD   PKTRAM
                031052  015320'                                                                 .WORD   RAMERR
1263
1264            031054                  40$:    ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                031054                                                          10000$:
                031054  104405                                                                  TRAP    C$ESEG
1265
```

```
 1266 031056  012364  000006                MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
 1267 031062  020327  003060'                CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
 1268 031066  103002                         BHIS    55$                     ;BRANCH IF ALL DATA DONE
 1269 031070  000137  030614'                JMP     5$                      ;BRANCH TILL BACK TO ZERO
 1270
 1271 031074                          55$:   ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      031074                                                                          L10056:
      031074  104403                                                                       TRAP    C$ESUB
 1272
 1273 031076  005737  002220'                TST     FATFLG                  ;ANY FATAL ERRORS ?
 1274 031102  001402                         BEQ     60$                     ;BRANCH IF NOT
 1275 031104  004737  017014'                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 1276 031110                          60$:
 1277
 1278                                        ;+
 1279                                        ;
 1280                                        ;TEST 7, SUBTEST 2
 1281                                        ;
 1282                                        ;CHECK THAT UNUSED BITS BEING SET CAUSES
 1283                                        ;WRITE CHARACTERISTICS COMMAND TO BE REJECTED
 1284                                        ;
 1285                                        ;-
 1286
 1287 031110                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      031110                                                                          T7.2:
      031110  104402                                                                  TRAP    C$BSUB
 1288
 1289 031112                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      031112  012700  000000                                                          MOV     #PRI00,R0
      031116  104441                                                                  TRAP    C$SPRI
 1290 031120  012703  032746'                MOV     #T72DATA,R3             ;START OF TEST DATA FOR SUBTEST
 1291 031124  012704  032700'          5$:   MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 1292 031130  004737  034062'                JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES
 1293
 1294 031134                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      031134  104404                                                                  TRAP    C$BSEG
 1295
 1296 031136  004737  015604'                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 1297 031142  103405                         BCS     10$                     ;BR IF SOFT INIT = OK
 1301 031144  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 1302 031146                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      031146  104455                                                                  TRAP    C$ERDF
      031150  001303                                                                  .WORD   707
      031152  003644'                                                                 .WORD   SFIERR
      031154  011644'                                                                 .WORD   SFIMSG
 1303 031156  005037  002222'          10$:   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 1304 031162  010400                         MOV     R4,R0                   ;START OF THE COMMAND PACKET
 1305 031164  061300                         ADD     (R3),R0                 ;OFFSET TO THE DATA WORD TO TEST
 1306 031166  056310  000002                BIS     2(R3),(R0)              ;SET THE DATA BITS TO BE TESTED
 1307 031172  010465  000000                MOV     R4,T5DB(R5)             ;SET THE PACKET ADDRESS
 1308 031176  004737  016060'                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 1309 031202  103405                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 1310 031204  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 1314 031206                                ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      031206  104455                                                                  TRAP    C$ERDF
      031210  001304                                                                  .WORD   708
      031212  033501'                                                                 .WORD   T7SSR
```

```
                031214  011656'                                                         .WORD   PKTSSR
1315            031216                  15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                031216  104406                                                          TRAP    C$CLP1
1316            031220                  ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
                031220  104410                                                          TRAP    C$ESCAPE
                031222  000116                                                          .WORD   10000$-.
1317            031224  005737  002222' TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1318            031230  001404          BEQ     22$                     ;BRANCH IF NOT
1322            031232                  ERRHRD  ERRNO,T7INT,PKTSSR
                031232  104456                                                          TRAP    C$ERHRD
                031234  001305                                                          .WORD   709
                031236  033661'                                                         .WORD   T7INT
                031240  011656'                                                         .WORD   PKTSSR
1323            031242  016501  000002  22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1324            031246  012702  102206          MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
1325            031252  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1326            031256  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1327            031260  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1328            031264  020201          25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1329            031266  001414                  BEQ     30$                     ;OKAY IF MATCH
1330            031270  010100                  MOV     R1,R0                   ;DATA FROM TSSR
1331            031272                          XOR     R2,R0                   ;FIND BITS IN ERROR
1332            031302  020027  002000          CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
1333            031306  001404                  BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1337            031310                          ERRHRD  ERRNO,T72REJ,PKTSSR     ;COMMAND NOT REJECTED
                031310  104456                                                          TRAP    C$ERHRD
                031312  001306                                                          .WORD   710
                031314  033113'                                                         .WORD   T72REJ
                031316  011656'                                                         .WORD   PKTSSR
1338            031320                  30$:    CKLOOP                  ;LOOP ON ERROR ?
                031320  104406                                                          TRAP    C$CLP1
1339            031322  032701  002000          BIT     #NBA,R1                 ;IS NBA BIT SET ?
1340            031326  001004                  BNE     35$                     ;OKAY IF NBA SET
1344            031330                          ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
                031330  104456                                                          TRAP    C$ERHRD
                031332  001307                                                          .WORD   711
                031334  032762'                                                         .WORD   T72NBA
                031336  011656'                                                         .WORD   PKTSSR
1345            031340                  35$:
1346            031340                          ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                031340                                                                  10000$:
                031340  104405                                                          TRAP    C$ESEG
1347
1348            031342  062703  000004          ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
1349            031346  020327  032762'         CMP     R3,#T72DONE             ;COMPARE TO END OF TEST DATA
1350            031352  103002                  BHIS    57$                     ;BRANCH IF ALL DATA TESTED
1351            031354  000137  031124'         JMP     5$                      ;BRANCH TILL BACK TO ZERO
1352
1353            031360                  57$:    ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
                031360                                                                  L10057:
                031360  104403                                                          TRAP    C$ESUB
1354
1355                                    ;*
1356                                    ;
1357                                    ;TEST 7, SUBTEST 3
1358                                    ;
1359                                    ;CHECK THE WRITE CHARACTERISTICS COMMAND IS REJECTED
```

```
                                        ;IF ISSUED WITH AN INVALID DATA BLOCK BYTE COUNT
                                        ;
                                        ;-
                                        
                        BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                T7.3:
031362  104402                                                  TRAP    C$BSUB
                                        
                        SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
031364  012700  000000                                          MOV     #PRI00,R0
031370  104441                                                  TRAP    C$SPRI
031372  012703  000001                MOV     #1,R3           ;STARTING BYTE COUNT
031376  012704  032700'        5$:     MOV     #T7PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
031402  004737  034062'                JSR     PC,T7REST       ;RESTORE PACKET TO STARTING VALUES
                                        
                        BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
031406  104404                                                  TRAP    C$BSEG
                                        
031410  004737  015604'                JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
031414  103405                        BCS     10$             ;BR IF SOFT INIT = OK
031416  010001                        MOV     R0,R1           ;SAVE CONTENTS OF TSSR
                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
031420  104455                                                  TRAP    C$ERDF
031422  001310                                                  .WORD   712
031424  003644'                                                 .WORD   SFIERR
031426  011644'                                                 .WORD   SFIMSG
031430  005037  002222'        10$:    CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
031434  010364  000006                MOV     R3,PKBCNT(R4)   ;INSERT THE BYTE COUNT FOR TEST
031440  010465  000000                MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
031444  004737  016060'                JSR     PC,WAITF        ;WAIT FOR SSR TO SET
031450  103405                        BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
031452  010001                        MOV     R0,R1           ;SAVE CONTENTS OF TSSR
                        ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
031454  104455                                                  TRAP    C$ERDF
031456  001311                                                  .WORD   713
031460  033501'                                                 .WORD   T7SSR
031462  011656'                                                 .WORD   PKTSSR
031464          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
031464  104406                                                  TRAP    C$CLP1
                        ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
031466  104410                                                  TRAP    C$ESCAPE
031470  000116                                                  .WORD   10000$-.
031472  005737  002222'                TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
031476  001404                        BEQ     22$             ;BRANCH IF NOT
                        ERRHRD  ERRNO,T7INT,PKTSSR
031500  104456                                                  TRAP    C$ERHRD
031502  001312                                                  .WORD   714
031504  033661'                                                 .WORD   T7INT
031506  011656'                                                 .WORD   PKTSSR
031510  016501  000002        22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
031514  012702  102206                MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
031520  032701  000100                BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
031524  001402                        BEQ     25$             ;BRANCH IF NOT OFF-LINE
031526  052702  000100                BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
031532  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
031534  001414                        BEQ     30$             ;OKAY IF MATCH
031536  010100                        MOV     R1,R0           ;DATA FROM TSSR
```

```
1498                                    ;+
1499                                    ;
1500                                    ;TEST 7, SUBTEST 5
1501                                    ;
1502                                    ;CHECK THAT WRITE CHARACTERISTICS COMMAND IS REJECTED IF THE
1503                                    ;MESSAGE BUFFER LENGTH SPECIFIES AN INVALID COUNT (LESS THAN 14)
1504                                    ;
1505                                    ;-
1506
1507 032054                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     032054                                                                                      T7.5:
     032054  104402                                                                                     TRAP    C$BSUB
1508
1509 032056                                     SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     032056  012700  000000                                                                             MOV     #PRI00,R0
     032062  104441                                                                                     TRAP    C$SPRI
1510 032064  012703  000001                     MOV     #1,R3                   ;STARTING BUFFER LENGTH
1511 032070  012704  032700'            5$:     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1512 032074  004737  034062'                    JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES
1513
1514 032100                                     BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     032100  104404                                                                                     TRAP    C$BSEG
1515
1516 032102  004737  015604'                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1517 032106  103405                            BCS     10$                     ;BR IF SOFT INIT = OK
1521 032110  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1522 032112                                     ERRDF   ERRN0,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     032112  104455                                                                                     TRAP    C$ERDF
     032114  001322                                                                                     .WORD   722
     032116  003644'                                                                                    .WORD   SFIERR
     032120  011644'                                                                                    .WORD   SFIMSG
1523 032122  005037  002222'        10$:        CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1524 032126  010337  032714'                    MOV     R3,T7DATA+4             ;INSERT THE BAD MESSAGE LENGTH
1525 032132  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1526 032136  004737  016060'                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1527 032142  103405                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1528 032144  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1532 032146                                     ERRDF   ERRN0,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     032146  104455                                                                                     TRAP    C$ERDF
     032150  001323                                                                                     .WORD   723
     032152  033501'                                                                                    .WORD   T7SSR
     032154  011656'                                                                                    .WORD   PKTSSR
1533 032156                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     032156  104406                                                                                     TRAP    C$CLP1
1534 032160                                     ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     032160  104410                                                                                     TRAP    C$ESCAPE
     032162  000116                                                                                     .WORD   10000$-.
1535 032164  005737  002222'                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1536 032170  001404                            BEQ     22$                     ;BRANCH IF NOT
1540 032172                                     ERRHRD  ERRN0,T7INT,PKTSSR
     032172  104456                                                                                     TRAP    C$ERHRD
     032174  001324                                                                                     .WORD   724
     032176  033661'                                                                                    .WORD   T7INT
     032200  011656'                                                                                    .WORD   PKTSSR
1541 032202  016501  000002        22$:        MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1542 032206  012702  102206                    MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
1543 032212  032701  000100                    BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

```
1544 032216  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1545 032220  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1546 032224  020201              25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1547 032226  001414                    BEQ     30$                     ;OKAY IF MATCH
1548 032230  010100                    MOV     R1,R0                   ;DATA FROM TSSR
1549 032232                            XOR     R2,R0                   ;FIND BITS IN ERROR
1550 032242  020027  002000            CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
1551 032246  001404                    BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1555 032250                            ERRHRD  ERRNO,T75REJ,PKTSSR     ;COMMAND NOT REJECTED
     032250  104456                                                            TRAP    C$ERHRD
     032252  001325                                                            .WORD   725
     032254  033403'                                                          .WORD   T75REJ
     032256  011656'                                                          .WORD   PKTSSR
1556 032260              30$:  CKLOOP                                  ;LOOP ON ERROR ?
     032260  104406                                                            TRAP    C$CLP1
1557 032262  032701  002000            BIT     #NBA,R1                 ;IS NBA BIT SET ?
1558 032266  001004                    BNE     35$                     ;OKAY IF NBA SET
1562 032270                            ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
     032270  104456                                                            TRAP    C$ERHRD
     032272  001326                                                            .WORD   726
     032274  032762'                                                          .WORD   T72NBA
     032276  011656'                                                          .WORD   PKTSSR
1563 032300              35$:
1564 032300                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     032300                                                                  10000$:
     032300  104405                                                            TRAP    C$ESEG
1565
1566 032302  005203                    INC     R3                      ;NEXT BUFFER LENGTH
1567 032304  020327  000016            CMP     R3,#14.                 ;HAVE ALL BAD VALUES BEEN TESTED ?
1568 032310  002002                    BGE     57$                     ;BRANCH IF ALL TESTED
1569 032312  000137  032070'          JMP     5$                      ;BRANCH TILL BACK TO ZERO
1570
1571 032316              57$:  ENDSUB                                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     032316                                                                  L10062:
     032316  104403                                                            TRAP    C$ESUB
1572
1573                           ;*
1574                           ;
1575                           ;TEST 7, SUBTEST 6
1576                           ;
1577                           ;THIS SUBTEST IS EXECUTED ONLY IF THE EXTENDED
1578                           ;FEATURES MODE IS ENABLED (AS DETERMINED BY EXAMINING
1579                           ;XST2 AFTER A PREVIOUS EXECUTION OF WRITE CHARACTERISTICS).
1580                           ;IT VERIFIES THAT A FIFTH CHARACTERISTICS DATA WORD IS FETCHED
1581                           ;IF THE BYTE COUNT PARAMETER IN THE COMMAND PACKET IS 10 DECIMAL
1582                           ;OR GREATER.
1583                           ;
1584                           ;-
1585 032320                            BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     032320                                                                  T7.6:
     032320  104402                                                            TRAP    C$BSUB
1586 032322  005737  002224'          TST     EXTFEA                  ;IS EXTENDED FEATURES SOFT. SW SET?
1587 032326  001002                    BNE     4$                      ;BR, IF SOFTWARE SWITCH IS SET (ON)
1588 032330                            EXIT    TST                     ;NO EXTENDED FEATURES EXIT THIS TEST
     032330  104432                                                            TRAP    C$EXIT
     032332  001576                                                            .WORD   L10055-.
1589 032334  004737  034062'    4$:   JSR     PC,T7REST               ;SET PACKET TO START-UP VALUES
```

```
1590
1591 032340                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     032340  012700  000000                                                                  MOV     #PRI00,R0
     032344  104441                                                                          TRAP    C$SPRI
1592 032346  012703  002764'                    MOV     #T$TBLK+12.,R3          ;START OF TEST DATA
1593 032352  012704  032700'                    MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1594 032356  012737  000200  032720'           MOV     #200,T7S?               ;SPECIAL BIT SET FOR EXTFEA RAM RD
1595 032364  012764  000012  000006           MOV     #10.,PKBCNT(R4)         ;START WITH EXTENDED FEATURES VALUE
1596 032372                            5$:
1597 032372                                    BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     032372  104404                                                                          TRAP    C$BSEG
1598
1599 032374  004737  015604'                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1600 032400  103405                            BCS     10$                     ;BR IF SOFT INIT = OK
1604 032402  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1605 032404                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     032404  104455                                                                          TRAP    C$ERDF
     032406  001327                                                                          .WORD   727
     032410  003644'                                                                         .WORD   SFIERR
     032412  011644'                                                                         .WORD   SFIMSG
1606 032414  005037  002220'           10$:     CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1607 032420  005037  002222'                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1608 032424  010465  000000                    MOV     R4,T$DB(R5)             ;SET THE PACKET ADDRESS
1609 032430  004737  016146'                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1610 032434  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1611 032436  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1615 032440                                    ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     032440  104455                                                                          TRAP    C$ERDF
     032442  001330                                                                          .WORD   728
     032444  033501'                                                                         .WORD   T7SSR
     032446  011656'                                                                         .WORD   PKTSSR
1616 032450  005237  002220'                    INC     FATFLG                  ;SET FATAL ERROR FLAG
1617 032454                            15$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     032454  104406                                                                          TRAP    C$CLP1
1618 032456                                    ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     032456  104410                                                                          TRAP    C$ESCAPE
     032460  000156                                                                          .WORD   10000$-.
1619 032462  005737  002222'                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1620 032466  001404                            BEQ     22$                     ;BRANCH IF NOT
1624 032470                                    ERRHRD  ERRNO,T7INT,PKTSSR
     032470  104456                                                                          TRAP    C$ERHRD
     032472  001331                                                                          .WORD   729
     032474  033661'                                                                         .WORD   T7INT
     032476  011656'                                                                         .WORD   PKTSSR
1625 032500  016501  000002           22$:     MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1626 032504  012702  000200                    MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
1627 032510  032701  000100                    BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1628 032514  001402                            BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1629 032516  052702  000100                    BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1630 032522  020201                    25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1631 032524  001404                            BEQ     30$                     ;OKAY IF MATCH
1635 032526                                    ERRHRD  ERRNO,T7NBA,PKTSSR      ;NBA NOT ZERO
     032526  104456                                                                          TRAP    C$ERHRD
     032530  001332                                                                          .WORD   730
     032532  033040'                                                                         .WORD   T7NBA
     032534  011656'                                                                         .WORD   PKTSSR
1636 032536                            30$:     CKLOOP                          ;LOOP ON ERROR ?
```

```
                032536  104406                                                          TRAP    C$CLP1
1637            032540  004737  016146'                 JSR     PC,CHKTSSR             ;WAIT FOR READY, NON-AMBIGUOUS
1638            032544  016501  000000                  MOV     TSBA(R5),R1            ;GET TSBA REGISTER CONTENTS
1639            032550  012702  032726'                 MOV     #T7BFR,R2              ;START OF THE DATA BUFFER
1640            032554  032762  000200  000012         BIT     #BIT7,XST2(R2)         ;IS EXTENDED FEATURES BIT SET ?
1641            032562  001402                          BEQ     32$                    ;BRANCH IF EXTENDED FEATURES OFF
1642            032564  062702  000002                  ADD     #2,R2                  ;EXTRA WORD IF SPECIAL FEATURES
1643            032570  062702  000016          32$:    ADD     #14.,R2                ;EXPECTED CONTENTS OF TSDA
1644            032574  020102                          CMP     R1,R2                  ;COMPARE EXPECTED TO RECEIVED
1645            032576  001404                          BEQ     35$                    ;ERROR IF NOT EQUAL
1649            032600                                  ERRHRD  ERRNO,T7TSBA,EXPREC    ;PRINT THE ERROR & EXPD/RECV
                032600  104456                                                          TRAP    C$ERHRD
                032602  001333                                                          .WORD   731
                032604  033750'                                                        .WORD   T7TSBA
                032606  015304'                                                        .WORD   EXPREC
1650
1651
1652            032610  012704  032710'          35$:    MOV     #T7DATA,R4             ;SET POINTER FOR CHECKER
1653            032614  004737  011034'                  JSR     PC,CKRAM2              ;SEE IF DATA IN RAM IS CORRECT
1654            032620  103404                          BCS     40$                    ;BRANCH IF PACKET IN RAM IS CORRECT
1658            032622                                  ERRHRD  ERRNO,PKTRAM,RAMERR    ;REPORT THE RAM ERROR(S)
                032622  104456                                                          TRAP    C$ERHRD
                032624  001334                                                          .WORD   732
                032626  004737'                                                        .WORD   PKTRAM
                032630  015320'                                                        .WORD   RAMERR
1659
1660            032632  012704  032700'          40$:    MOV     #T7PACKET,R4           ;RESET PACKET POINTER
1661            032636                                  ENDSEG                         ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                032636                                                                      10000$:
                032636  104405                                                                  TRAP    C$ESEG
1662
1663            032640  012364  000006                  MOV     (R3)+,PKBCNT(R4)       ;SET THE TEST WORD
1664            032644  020327  003060'                 CMP     R3,#TBLEND             ;HAS ALL DATA BEEN TESTED ?
1665            032650  103002                          BHIS    55$                    ;BRANCH IF ALL DATA DONE
1666            032652  000137  032372'                 JMP     5$                     ;BRANCH TILL BACK TO ZERO
1667
1668            032656                          55$:    ENDSUB                         ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
                032656                                                                      L10063:
                032656  104403                                                                  TRAP    C$ESUB
1675
                032660  005737  002220'                 TST     FATFLG                 ;ANY FATAL ERRORS ?
                032664  001402                          BEQ     60$                    ;BRANCH IF NOT
                032666  004737  017014'                 JSR     PC,CKDROP              ;TRY TO DROP THE UNIT
                032672                          60$:
                032672                                  EXIT    TST                    ;ALL DONE THIS TEST
                032672  104432                                                          TRAP    C$EXIT
                032674  001234                                                          .WORD   L10055-.
1677
1678                                            ;+
1679                                            ;LOCAL STORAGE FOR THIS TEST
1680                                            ;-
1681
1682            032676                                  .BLKB   10-<.-TSV2&7>
1683            032700                          T7PACKET:
1684            032700  100004                          .WORD   100004                 ;COMMAND PACKET FOR TEST
1685            032702  032710'                         .WORD   T7DATA                 ;WRITE CHARACTERISTICS COMMAND, WITH ACK
1686            032704  000000                          .WORD   0                      ;ADDRESS OF CHARACTERISTICS BLOCK
```

```
1687 032706  000010                        .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
1688
1689 032710                          T7DATA:                               ;CHARACTERISTICS DATA BLOCK
1690 032710  032726'                        .WORD   T7BFR                   ;ADDRESS OF MESSAGE BUFFER
1691 032712  000000                        .WORD   0
1692 032714  000016                        .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
1693 032716  000000                        .WORD   0
1694 032720  000000                  T7SP: .WORD   0                       ;EXTFEA EXTRA WORD
1695
1696 032722  000000  000000               .WORD   0,0                     ;SPACE
1697 032726                          T7BFR: .BLKW  8.                      ;MESSAGE BUFFER
1698
1699                                 ;+
1700                                 ;
1701                                 ;TEST DATA FOR SUBTEST TWO
1702                                 ;
1703                                 ;DATA HAS FORMAT:
1704                                 ;
1705                                 ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
1706                                 ;       2ND WORD        BITS TO SET FOR TEST
1707                                 ;
1708                                 ;-
1709
1710 032746                          T72DATA:
1711 032746  000000  037140                .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
1712 032752  000002  000001                .WORD   2,BIT0
1713 032756  000004  100100                .WORD   4,BIT6!BIT15
1714         032762'                 T72DONE=.
1715
1716
1717                                 ;+
1718                                 ;LOCAL TEXT MESSAGES FOR TEST
1719                                 ;-
1720
1721 032762     116     102     101 T72NBA: .ASCIZ  'NBA Not Set On Rejected WRITE CHARACTERISTICS'
1722 033040     127     122     111 T7NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
1723 033113     127     122     111 T72REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
1724 033212     127     122     111 T73REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
1725 033305     127     122     111 T74REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
1726 033403     127     122     111 T75REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
1727 033501     103     157     156 T7SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
1728 033570     105     170     160 T7NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
1729 033661     125     156     145 T7INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
1730 033750     111     156     143 T7TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
1731 034033     127     162     151 TST7ID: .ASCIZ  'Write Characteristics'
1732                                        .EVEN
1733
1734
1735                                 ;+
1736                                 ;
1737                                 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1738                                 ;
1739                                 ;-
1740
1741 034062                          T7REST:
1742 034062                                 SAVREG                          ;SAVE THE REGISTERS
1743 034066  012701  032700'               MOV     #T7PACKET,R1            ;START OF THE PACKET
```

```
1744 034072  012721  100004          MOV     #100004,(R1)+   ;WRITE CHARACTERISTICS WITH ACK
1745 034076  012721  032710'        MOV     #T7DATA,(R1)+   ;ADDRESS OF CHAR DATA BLOCK
1746 034102  005021                 CLR     (R1)+           ;EXTENDED ADDRESS
1747 034104  012721  000010         MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
1748 034110  012721  032726'        MOV     #T7BFR,(R1)+    ;ADDRESS OF MESSAGE BUFFER
1749 034114  005021                 CLR     (R1)+
1750 034116  012721  000020         MOV     #16.,(R1)+      ;LENGTH OF MESSAGE BUFFER
1751 034122  005021                 CLR     (R1)+
1752 034124  005011                 CLR     (R1)
1753 034126  000207                 RTS     PC              ;RETURN
1754 034130                         ENDTST
     034130                                                                 L10055:
     034130  104401                                                                TRAP    C$ETST
1755
1756
1757                                .SBTTL  TEST  8: VOLUME CHECK
1758
1759
1760                        ;       THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD
1761                        ;       WITHIN THE M7455 AND APPEARING IN XSTO, IS SET BY INITIALIZE AND
1762                        ;       CLEARED BY EXECUTING A WRITE CHARACTERISTICS COMMAND WITH THE
1763                        ;       CVC BIT SET.  IT IS ALSO VERIFIED THAT A WRITE CHARACTERISTICS
1764                        ;       COMMAND WITH THE CVC BIT CLEAR DOES NOT AFFECT THE STATE OF THE
1765                        ;       VOLUME CHECK BIT.  THE ACTUAL FUNCTION OF VOLUME CHECK, THAT OF
1766                        ;       PREVENTING OR ALLOWING A TAPE MOTION COMMAND DEPENDING UPON
1767                        ;       WHETHER VOLUME CHECK IS SET OR CLEAR, IS NOT CHECKED BY THIS
1768                        ;       TEST;  THIS FUNCTIONALITY IS CHECKED IN THE INDIVIDUAL TESTS OF
1769                        ;       TAPE MOTION COMMANDS.
1770                        ;
1771                        ;       THE TEST PROCEEDS AS FOLLOWS:
1772                        ;
1773                        ;            1.  THE CONTROLLER IS INITIALIZED BY WRITING INTO THE TSSR.
1774                        ;
1775                        ;            2.  A WRITE CHARACTERISTICS COMMAND IS ISSUED (WITH CVC=0)
1776                        ;                AND XSTO IN THE RETURNED MESSAGE BUFFER IS EXAMINED;
1777                        ;                THE VCK BIT SHOULD BE CLEAR (0).
1778                        ;
1779                        ;            3.  THE PREVIOUS STEP IS REPEATED TO VERIFY THAT VCK DOES
1780                        ;                NOT CHANGE (REMAINS AT 0).
1781                        ;
1782                        ;            4.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=1
1783                        ;                AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
1784                        ;                EXAMINED;  THE VCK BIT SHOULD BE CLEAR (0).
1785                        ;
1786                        ;            5.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=0
1787                        ;                AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
1788                        ;                EXAMINED;  THE VCK BIT SHOULD REMAIN CLEAR (0).
1789                        ;
1790
1791 034132                         BGNTST
     034132                                                                 T8::
1796 034132  012700  035017'        MOV     #TST8ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
1797 034136  004737  016322'        JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
1798 034142  012737  000024  002214'  MOV    #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
1799 034150                         T8LOOP:
1800
1801 034150  012704  034540'        MOV     #T8PACKET,R4            ;PACKET FOR WRITE CHARACTERISTICS
```

```
1802 034154  004737  015604'          5$:    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1803 034160  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
1807 034162  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1808 034164                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     034164  104455                                                                  TRAP    C$ERDF
     034166  001441                                                                  .WORD   801
     034170  003644'                                                                 .WORD   SFIERR
     034172  011644'                                                                 .WORD   SFIMSG
1809 034174  042714  040000          10$:    BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
1810 034200  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1811 034204  004737  016146'                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1812 034210  103405                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1813 034212  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1817 034214                                  ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     034214  104455                                                                  TRAP    C$ERDF
     034216  001442                                                                  .WORD   802
     034220  034730'                                                                 .WORD   T8SSR
     034222  011656'                                                                 .WORD   PKTSSR
1818 034224                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034224  104406                                                                  TRAP    C$CLP1
1819 034226                                  ESCAPE  TST                     ;EXIT IF FATAL ERROR
     034226  104410                                                                  TRAP    C$ESCAPE
     034230  000604                                                                  .WORD   L10064-.
1820 034232  012702  034562'                MOV     #T8BFR,R2               ;ADDRESS OF THE MESSAGE BUFFER
1821 034236  032762  000020  000006        BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK SET IN XSTO ?
1822 034244  001406                          BEQ     20$                     ;OKAY IF VOLUME CHECK IS CLEAR
1826 034246  016501  000002                  MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
1827 034252                                  ERRHRD  ERRNO,T8NVCK,PKTMES     ;VOLUME CHECK NOT CLEAR
     034252  104456                                                                  TRAP    C$ERHRD
     034254  001443                                                                  .WORD   803
     034256  034637'                                                                 .WORD   T8NVCK
     034260  011720'                                                                 .WORD   PKTMES
1828 034262                          20$:    CKLOOP                          ;LOOP ON ERROR ?
     034262  104406                                                                  TRAP    C$CLP1
1829 034264  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1830 034270  004737  016146'                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1831 034274  103405                          BCS     25$                     ;BR IF CARRY SET (GOOD RETURN)
1832 034276  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1836 034300                                  ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     034300  104455                                                                  TRAP    C$ERDF
     034302  001444                                                                  .WORD   804
     034304  034730'                                                                 .WORD   T8SSR
     034306  011656'                                                                 .WORD   PKTSSR
1837 034310                          25$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034310  104406                                                                  TRAP    C$CLP1
1838 034312                                  ESCAPE  TST                     ;EXIT IF FATAL ERROR
     034312  104410                                                                  TRAP    C$ESCAPE
     034314  000520                                                                  .WORD   L10064-.
1839 034316  032762  000020  000006        BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK SET IN XSTO ?
1840 034324  001406                          BEQ     30$                     ;OKAY IF VOLUME CHECK IS SET
1844 034326  016501  000002                  MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
1845 034332                                  ERRHRD  ERRNO,T8NVCK,PKTMES     ;VOLUME CHECK NOT SET
     034332  104456                                                                  TRAP    C$ERHRD
     034334  001445                                                                  .WORD   805
     034336  034637'                                                                 .WORD   T8NVCK
     034340  011720'                                                                 .WORD   PKTMES
1846 034342                          30$:    CKLOOP                          ;LOOP ON ERROR ?
```

```
                034342  104406                                                          TRAP    C$CLP1
1847            034344  052714  040000                  BIS     #BIT14,(R4)             ;SET THE CVC BIT
1848            034350  010465  000000                  MOV     R4,T5DB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1849            034354  004737  016146'                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1850            034360  103405                          BCS     35$                     ;BR IF CARRY SET (GOOD RETURN)
1851            034362  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1855            034364                                  ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                034364  104455                                                          TRAP    C$ERDF
                034366  001446                                                          .WORD   806
                034370  034730'                                                        .WORD   T8SSR
                034372  011656'                                                        .WORD   PKTSSR
1856            034374                          35$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                034374  104406                                                          TRAP    C$CLP1
1857            034376                                  ESCAPE  TST                     ;EXIT IF FATAL ERROR
                034376  104410                                                          TRAP    C$ESCAPE
                034400  000434                                                          .WORD   L10064-.
1858            034402  032762  000020  000006          BIT     #XSOVCK,XST0(R2)        ;IS VOLUME CHECK CLEAR IN XST0 ?
1859            034410  001406                          BEQ     40$                     ;OKAY IF VOLUME CHECK IS CLEARED
1863            034412  016501  000002                  MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
1864            034416                                  ERRHRD  ERRNO,T8VCK,PKTMES      ;VOLUME CHECK NOT CLEARED
                034416  104456                                                          TRAP    C$ERHRD
                034420  001447                                                          .WORD   807
                034422  034602'                                                        .WORD   T8VCK
                034424  011720'                                                        .WORD   PKTMES
1865            034426                          40$:    CKLOOP                          ;LOOP ON ERROR ?
                034426  104406                                                          TRAP    C$CLP1
1866            034430  042714  040000                  BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
1867            034434  010465  000000                  MOV     R4,T5DB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1868            034440  004737  016146'                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1869            034444  103405                          BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
1870            034446  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1874            034450                                  ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                034450  104455                                                          TRAP    C$ERDF
                034452  001450                                                          .WORD   808
                034454  034730'                                                        .WORD   T8SSR
                034456  011656'                                                        .WORD   PKTSSR
1875            034460                          45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                034460  104406                                                          TRAP    C$CLP1
1876            034462                                  ESCAPE  TST                     ;EXIT IF FATAL ERROR
                034462  104410                                                          TRAP    C$ESCAPE
                034464  000350                                                          .WORD   L10064-.
1877            034466  032762  000020  000006          BIT     #XSOVCK,XST0(R2)        ;IS VOLUME CHECK CLEAR IN XST0 ?
1878            034474  001406                          BEQ     50$                     ;OKAY IF VOLUME CHECK IS CLEARED
1882            034476  016501  000002                  MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
1883            034502                                  ERRHRD  ERRNO,T8VCK,PKTMES      ;VOLUME CHECK NOT CLEARED
                034502  104456                                                          TRAP    C$ERHRD
                034504  001451                                                          .WORD   809
                034506  034602'                                                        .WORD   T8VCK
                034510  011720'                                                        .WORD   PKTMES
1884            034512                          50$:    CKLOOP                          ;LOOP ON ERROR ?
                034512  104406                                                          TRAP    C$CLP1
1885            034514  004737  016270'         60$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
1886            034520  103002                          BCC     62$                     ;BRANCH IF NOT
1887            034522  000137  034150'                 JMP     T8LOOP                  ;LOOP UNTIL COUNT EXPIRED
1888            034526                          62$:    EXIT    TST                     ;ALL DONE THIS TEST
                034526  104432                                                          TRAP    C$EXIT
                034530  000304                                                          .WORD   L10064-.
```

```
1889
1890                                    ;+
1891                                    ;LOCAL STORAGE FOR THIS TEST
1892                                    ;-
1893
1895 034532                                     .BLKB   10-<.-TSV2&7>
1897 034540                             T8PACKET:                       ;COMMAND PACKET FOR TEST
1898 034540  100004                             .WORD   100004          ;WRITE CHARACTERISTICS COMMAND
1899 034542  034550'                            .WORD   T8DATA          ;ADDRESS OF CHARACTERISTICS BLOCK
1900 034544  000000                             .WORD   0
1901 034546  000010                             .WORD   10              ;STARTING VALUE OF COUNTER
1902
1903 034550                             T8DATA:                         ;CHARACTERISTICS DATA BLOCK
1904 034550  034562'                            .WORD   T8BFR           ;ADDRESS OF MESSAGE BUFFER
1905 034552  000000                             .WORD   0
1906 034554  000020                             .WORD   16.             ;LENGTH OF MESSAGE BUFFER
1907 034556  000000  000000                      .WORD   0,0
1908
1909 034562                             T8BFR:  .BLKW   8.              ;MESSAGE BUFFER
1910
1911
1912                                    ;+
1913                                    ;LOCAL TEXT MESSAGES FOR TEST
1914                                    ;-
1915
1916 034602    126     157     154     T8VCK:  .ASCIZ  'Volume Check Bit Not Cleared'
1917 034637    126     157     154     T8NVCK: .ASCIZ  'Volume Check Bit (VCK) Not Clear After Initialize (XSTO)'
1918 034730    103     157     156     T8SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Characteristics'
1919 035017    126     157     154     TST8ID: .ASCIZ  'Volume Check'
1920                                            .EVEN
1921 035034                             ENDTST
     035034                                                     L10064:
     035034  104401                                                      TRAP    C$ETST
1922
1923                                    .SBTTL  TEST  9: COMPLETION INTERRUPT
1924
1925
1926                                    ;       THIS TEST VERIFIES THAT AN INTERRUPT IS GENERATED AT THE
1927                                    ;       COMPLETION OF THE WRITE CHARACTERISTICS COMMAND IF THE INTERRUPT
1928                                    ;       ENABLE (IE) BIT IN THE COMMAND HEADER WORD IS SET.  THIS TEST
1929                                    ;       CHECKS THE FUNCTIONING OF THE INTERRUPT LOGIC AND BASIC
1930                                    ;       PROCESSING OF THE IE BIT.
1931                                    ;
1932                                    ;       THE SEQUENCES OF TEST 7 ARE REPEATED, EXCEPT THAT THE INTERRUPT
1933                                    ;       SERVICE ROUTINE IS SET UP TO EXPECT INTERRUPTS AND EACH WRITE
1934                                    ;       CHARACTERISTICS COMMAND IS ISSUED WITH THE IE BIT SET (1).  IT
1935                                    ;       IS VERIFIED, WHERE APPROPRIATE, THAT THE IE STATUS BIT IN XSTO
1936                                    ;       OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETION INTERRUPT IS
1937                                    ;       GENERATED.  FINALLY, A SEQUENCE OF TWO COMMANDS ARE ISSUED, THE
1938                                    ;       FIRST WITH IE=1 AND THE SECOND WITH IE=0.  IT IS VERIFIED THAT
1939                                    ;       NO INTERRUPT IS GENERATED AFTER THE SECOND COMMAND AND THAT THE
1940                                    ;       IE BIT IN XSTO IS 0.
1941                                    ;
1942 035036                             BGNTS1
     035036                                                             T9::
1947 035036  005037  002224'                    CLR     EXTFEA          ;CLEAR EXTENDED FEATURES SWITCH
1948 035042  012700  040061'                    MOV     #TST9ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
```

```
 1949 035046  004737  016322'                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
 1950 035052  012737  000024  002214'            MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
 1951 035060                              T9LOOP:
 1952 035060                                      BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      035060                                                                              T9.1:
      035060  104402                                                                          TRAP    C$BSUB
 1953
 1954 035062  004737  040106'                    JSR     PC,T9REST               ;SET PACKET TO INITIAL VALUES
 1955 035066                                      SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      035066  012700  000000                                                              MOV     #PRI00,R0
      035072  104441                                                                      TRAP    C$SPRI
 1956 035074  012703  002762'                    MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
 1957 035100  012704  037010'                    MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 1958 035104  012764  000010  000006            MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
 1959 035112                              5$:
 1960 035112                                      BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      035112  104404                                                                      TRAP    C$BSEG
 1961
 1962 035114  004737  015604'                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 1963 035120  103405                              BCS     10$                     ;BR IF SOFT INIT = OK
 1967 035122  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 1968 035124                                      ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      035124  104455                                                                      TRAP    C$ERDF
      035126  001605                                                                      .WORD   9C1
      035130  003644'                                                                     .WORD   SFIERR
      035132  011644'                                                                     .WORD   SFIMSG
 1969 035134  005037  002220'              10$:   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 1970 035140  005037  002222'                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 1971 035144  010465  000000                      MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 1972 035150  004737  016146'                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 1973 035154  103407                              BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 1974 035156  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 1978 035160                                      ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      035160  104455                                                                      TRAP    C$ERDF
      035162  001606                                                                      .WORD   902
      035164  037527'                                                                     .WORD   T9SSR
      035166  011656'                                                                     .WORD   PKTSSR
 1979 035170  005237  002220'                    INC     FATFLG                  ;SET FATAL ERROR FLAG
 1980 035174                              15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      035174  104406                                                                      TRAP    C$CLP1
 1981 035176                                      ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      035176  104410                                                                      TRAP    C$ESCAPE
      035200  000056                                                                      .WORD   10000$-.
 1982 035202  005737  002222'                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 1983 035206  001004                              BNE     22$                     ;BRANCH IF YES
 1987 035210                                      ERRHRD  ERRNO,T9NINT,PKTSSR
      035210  104456                                                                      TRAP    C$ERHRD
      035212  001607                                                                      .WORD   903
      035214  037616'                                                                     .WORD   T9NINT
      035216  011656'                                                                     .WORD   PKTSSR
 1988 035220  016501  000002              22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 1989 035224  012702  000200                      MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 1990 035230  032701  000100                      BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 1991 035234  001402                              BEQ     25$                     ;BRANCH IF NOT OFF LINE
 1992 035236  052702  000100                      BIS     #OFL,R2                 ;SET OFF LINE IN EXPECTED DATA
 1993 035242  020201                      25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 1994 035244  001404                              BEQ     30$                     ;OKAY IF MATCH
```

```
1998 035246                                ERRHRD  ERRNO,T9NBA,PKTSSR      ;NBA NOT ZERO
     035246  104456                                                                TRAP    C$ERHRD
     035250  001610                                                                .WORD   904
     035252  037066'                                                               .WORD   T9NBA
     035254  011656'                                                               .WORD   PKTSSR
1999 035256                          30$:
2000 035256                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     035256                                                                      10000$:
     035256  104405                                                                TRAP    C$ESEG
2001
2002 035260  012364  000006                MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
2003 035264  020327  003060'                CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
2004 035270  103002                         BHIS    55$                     ;BRANCH IF ALL DATA DONE
2005 035272  000137  035112'                JMP     5$                      ;BRANCH TILL BACK TO ZERO
2006
2007 035276                          55$:   ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     035276                                                                      L10066:
     035276  104403                                                                TRAP    C$ESUB
2008
2009 035300  005737  002220'                TST     FATFLG                  ;ANY FATAL ERRORS ?
2010 035304  001402                         BEQ     60$                     ;BRANCH IF NOT
2011 035306  004737  017014'                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2012 035312  033727  037044'  000200  60$:  BIT     T9BFR+12,#BIT7          ;EXTENDED FEATURES SET?
2013 035320  001402                         BEQ     70$                     ;BR IF NO
2014 035322  005237  002224'                INC     EXTFEA                  ;SET EXT FEATURE FLAG
2015 035326                          70$:
2016
2017                                       ;+
2018                                       ;
2019                                       ;TEST 9, SUBTEST 2
2020                                       ;
2021                                       ;CHECK THAT UNUSED BITS BEING SET CAUSES
2022                                       ;WRITE CHARACTERISTICS COMMAND TO BE REJECTED
2023                                       ;
2024                                       ;-
2025
2026 035326                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     035326                                                                      T9.2:
     035326  104402                                                                TRAP    C$BSUB
2027
2028 035330                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     035330  012700  000000                                                         MOV     #PRI00,R0
     035334  104441                                                                TRAP    C$SPRT
2029 035336  012703  037052'                MOV     #T92DATA,R3             ;START OF TEST DATA FOR SUBTEST
2030 035342  012704  037010'          5$:   MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2031 035346  004737  040106'                JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES
2032
2033 035352                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     035352  104404                                                                TRAP    C$BSEG
2034
2035 035354  004737  015604'                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2036 035360  103405                         BCS     10$                     ;BR IF SOFT INIT = OK
2040 035362  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF ISSR
2041 035364                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     035364  104455                                                                TRAP    C$ERDF
     035366  001611                                                                .WORD   905
     035370  003644'                                                               .WORD   SFIERR
```

```
                020156  103352                          BCC     1$
2742            020160  013700  020316'                 MOV     MENRES,R0       ;GET THE OPERATOR'S REPLY
2743            020164  020001                          CMP     R0,R1           ;COMPARE TO MAXIMUM ALLOWED
2744            020166  101411                          BLOS    5$              ;BRANCH IF OK
2745            020170                                  PRINTF  #MENERR         ;DISPLAY ERROR MESSAGE
                020170  012746  020214'                 MOV     #MENERR,-(SP)
                020174  012746  000001                  MOV     #1,-(SP)
                020200  010600                          MOV     SP,R0
                020202  104417                          TRAP    C$PNTF
                020204  062706  000004                  ADD     #4,SP
2746            020210  000735                          BR      1$              ;RETRY
2747            020212  000207                  5$:     RTS     PC              ;RETURN TO CALLER
2748            020214     045     116     045  MENERR: .ASCIZ  '<N><A *** Menu Selection Too Large ***'
2749            020262     045     116     045  SELASC: .ASCIZ  '<N><T'
2750            020267     105     156     164  MENASC: .ASCIZ  'Enter Menu Selection: '
2751                                                    .EVEN
2752            020316  000000                  MENRES:         .WORD   0
2753
2754                                                    .SBTTL  CHKMAN  - CHECK MANUAL INTERVENTION LEGALITY
2755                                            ;+
2756                                            ;
2757                                            ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2758                                            ;
2759                                            ;INPUT:
2760                                            ;
2761                                            ;       NONE.
2762                                            ;
2763                                            ;OUTPUT:
2764                                            ;
2765                                            ;       CARRY   0       MANUAL INTERVENTION NOT ALLOWED
2766                                            ;               1       MANUAL INTERVENTION IS OK
2767                                            ;
2768                                            ;SIDE EFFECTS:
2769                                            ;
2770                                            ;       A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2771                                            ;       NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2772                                            ;       ALLOWED.
2773                                            ;
2774                                            ;-
2775
2776            020320                          CHKMAN::
2777            020320                                  SAVREG                  ;SAVE THE REGISTERS
2778            020324                                  MANUAL                  ;SEE IF MANUAL INTERVENTION OK
                020324  104450                          TRAP    C$MANI
2779            020326                                  BCOMPLETE 1$            ;BRANCH IF ALLOWED
                020326  103411                          BCS     1$
2780            020330                                  PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
                020330  012746  020354'                 MOV     #NOMAN,-(SP)
                020334  012746  000001                  MOV     #1,-(SP)
                020340  010600                          MOV     SP,R0
                020342  104417                          TRAP    C$PNTF
                020344  062706  000004                  ADD     #4,SP
2781            020350  000241                          CLC                     ;CLEAR CARRY FOR ERROR
2782            020352  000207                  1$:     RTS     PC              ;RETURN
2783
2784            020354     045     116     045  NOMAN:  .ASCIZ  '<N><A *** Manual Intervention not Allowed - Test Aborted ***'
2785                                                    .even
```

```
                                                                                        TRAP    C$ERHRD
                                                                                        .WORD   912
                                                                                        .WORD   T93REJ
                                                                                        .WORD   PKTSSR
2137                            30$:
2138                                    ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                                                                                      10000$:
                                                                                        TRAP    C$ESEG
2139
2140                                    INC     R3                      ;NEXT BYTE COUNT
2141                                    CMP     R3,#6                   ;TESTED ALL INVALID ?
2142                                    BGE     59$                     ;BRANCH IF TEST DONE
2143                                    JMP     5$                      ;BRANCH TILL BACK TO ZERO
2144
2145                            59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                                                                                      L10070:
                                                                                        TRAP    C$ESUB
2146
2147                            ;+
2148                            ;
2149                            ;TEST 9, SUBTEST 4
2150                            ;
2151                            ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
2152                            ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
2153                            ;
2154                            ;-
2155
2156                                    BGNSUB                          ;/////////////// BEGIN SUBTEST ////////////
                                                                                        T9.4:
                                                                                        TRAP    C$BSUB
2157
2158                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                                        MOV     #PRI00,R0
                                                                                        TRAP    C$SPRI
2159                            5$:     MOV     #T92DATA,R3             ;START OF TEST DATA FOR SUBTEST
2160                                    MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2161                                    JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES
2162
2163
2164                                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2165                                    BCS     10$                     ;BR IF SOFT INIT = OK
2169                                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2170                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   913
                                                                                        .WORD   SFIERR
                                                                                        .WORD   SFIMSG
2171                            10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2172                                    BIS     #1,T9DATA               ;MAKE ADDRESS ODD
2173                                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2174                                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
2175                                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2176                                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2180                                    ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   914
                                                                                        .WORD   T9SSR
```

```
     036044  011656'                                                              .WORD    PKTSSR
2181 036046                    15$:    CKLOOP                       ;LOOP ON ERROR, IF FLAG SET
     036046  104406                                                               TRAP     C$CLP1
2182 036050                            ESCAPE  SUB                  ;BY-PASS SUBTEST IF FATAL ERROR
     036050  104410                                                               TRAP     C$ESCAPE
     036052  000056                                                               .WORD    L10071-.
2183 036054  005737  002222'           TST     INTRECV              ;DID AN INTERRUPT OCCUR ?
2184 036060  001004                    BNE     22$                  ;BRANCH IF YES
2188 036062                            ERRHRD  ERRNO,T9NINT,PKTSSR
     036062  104456                                                               TRAP     C$ERHRD
     036064  001623                                                               .WORD    915
     036066  037616'                                                              .WORD    T9NINT
     036070  011656'                                                              .WORD    PKTSSR
2189 036072  016501  000002    22$:    MOV     TSSR(R5),R1          ;GET THE CONTENTS OF TSSR
2190 036076  012702  102206            MOV     #SC!SSR!TSREJ!NBA,R2 ;EXPECTED CONTENTS OF TSSR
2191 036102  032701  000100            BIT     #OFL,R1              ;IS OFF-LINE BIT SET ?
2192 036106  001402                    BEQ     25$                  ;BRANCH IF NOT OFF-LINE
2193 036110  052702  000100            BIS     #OFL,R2              ;SET OFF-LINE IN EXPECTED DATA
2194 036114  020201            25$:    CMP     R2,R1                ;DOES EXPECTED MATCH RECEIVED ?
2195 036116  001404                    BEQ     30$                  ;OKAY IF MATCH
2199 036120                            ERRHRD  ERRNO,T94REJ,PKTSSR  ;COMMAND NOT REJECTED
     036120  104456                                                               TRAP     C$ERHRD
     036122  001624                                                               .WORD    916
     036124  037333'                                                              .WORD    T94REJ
     036126  011656'                                                              .WORD    PKTSSR
2200 036130                    30$:
2201
2202 036130                            ENDSUB                       ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     036130                                                               L10071:
     036130  104403                                                               TRAP     C$ESUB
2203
2204
2205                           ;+
2206                           ;
2207                           ;TEST 9, SUBTEST 5
2208                           ;
2209                           ;CHECK THAT WRITE CHARACTERISTICS COMMAND IS REJECTED IF THE
2210                           ;MESSAGE BUFFER LENGTH SPECIFIES AN INVALID COUNT (LESS THAN 14)
2211                           ;
2212                           ;-
2213
2214 036132                            BGNSUB                       ;////////////// BEGIN SUBTEST ////////////
     036132                                                               T9.5:
     036132  104402                                                               TRAP     C$BSUB
2215
2216 036134                            SETPRI  #PRI00               ;LOWER PRIORITY TO ALLOW INTERRUPTS
     036134  012700  000000                                                       MOV      #PRI00,R0
     036140  104441                                                               TRAP     C$SPRI
2217 036142  012703  000001            MOV     #1,R3                ;STARTING BUFFER LENGTH
2218 036146  012704  037010'   5$:     MOV     #T9PACKET,R4         ;GET THE ADDRESS OF COMMAND PACKET
2219 036152  004737  040106'           JSR     PC,T9REST            ;RESTORE PACKET TO STARTING VALUES
2220
2221 036156                            BGNSEG                       ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     036156  104404                                                               TRAP     C$BSEG
2222
2223 036160  004737  015604'           JSR     PC,SOFINIT           ;DO SOFT INIT OF CONTROLLER
2224 036164  103405                    BCS     10$                  ;BR IF SOFT INIT = OK
```

```
2228 036166  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2229 036170                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     036170  104455                                                                 TRAP    C$ERDF
     036172  001625                                                                 .WORD   917
     036174  003644'                                                                .WORD   SFIERR
     036176  011644'                                                                .WORD   SFIMSG
2230 036200  005037  002222'          10$:   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2231 036204  010337  037024'                 MOV     R3,T9DATA+4             ;INSERT THE BAD MESSAGE LENGTH
2232 036210  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2233 036214  004737  016060'                 JSR     PC,WAITF                ;WAIT FOR SSR TO SET
2234 036220  103405                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2235 036222  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2239 036224                                 ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     036224  104455                                                                 TRAP    C$ERDF
     036226  001626                                                                 .WORD   918
     036230  037527'                                                                .WORD   T9SSR
     036232  011656'                                                                .WORD   PKTSSR
2240 036234                         15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036234  104406                                                                 TRAP    C$CLP1
2241 036236                                 ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     036236  104410                                                                 TRAP    C$ESCAPE
     036240  000056                                                                 .WORD   10000$-.
2242 036242  005737  002222'                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2243 036246  001004                         BNE     22$                     ;BRANCH IF YES
2247 036250                                 ERRHRD  ERRNO,T9NINT,PKTSSR
     036250  104456                                                                 TRAP    C$ERHRD
     036252  001627                                                                 .WORD   919
     036254  037616'                                                                .WORD   T9NINT
     036256  011656'                                                                .WORD   PKTSSR
2248 036260  016501  000002          22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2249 036264  012702  102206                 MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
2250 036270  032701  000100                 BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2251 036274  001402                         BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2252 036276  052702  000100                 BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2253 036302  020201                 25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2254 036304  001404                         BEQ     30$                     ;OKAY IF MATCH
2258 036306                                 ERRHRD  ERRNO,T95REJ,PKTSSR     ;COMMAND NOT REJECTED
     036306  104456                                                                 TRAP    C$ERHRD
     036310  001630                                                                 .WORD   920
     036312  037431'                                                                .WORD   T95REJ
     036314  011656'                                                                .WORD   PKTSSR
2259 036316                         30$:
2260 036316                                 ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     036316                                                                     10000$:
     036316  104405                                                                 TRAP    C$ESEG
2261
2262 036320  005203                         INC     R3                      ;NEXT BUFFER LENGTH
2263 036322  020327  000014                 CMP     R3,#14                  ;HAVE ALL BAD VALUES BEEN TESTED ?
2264 036326  002002                         BGE     59$                     ;BRANCH IF ALL TESTED
2265 036330  000137  036146'                JMP     5$                      ;BRANCH TILL BACK TO ZERO
2266
2267 036334                         59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     036334                                                                     L10072:
     036334  104403                                                                 TRAP    C$ESUB
2268                                ;*
2269                                ;
2270                                ;TEST 9, SUBTEST 6
```

```
2271                                    ;
2272                                    ;THIS SUBTEST IS EXECUTED ONLY IF THE EXTENDED
2273                                    ;FEATURES MODE IS ENABLED (AS DETERMINED BY EXAMINING
2274                                    ;XST2 AFTER A PREVIOUS EXECUTION OF WRITE CHARACTERISTICS).
2275                                    ;IT VERIFIES THAT A FIFTH CHARACTERISTICS DATA WORD IS FETCHED
2276                                    ;IF THE BYTE COUNT PARAMETER IN THE COMMAND PACKET IS 10 DECIMAL
2277                                    ;OR GREATER.
2278                                    ;
2279                                    ;-
2280 036336                                    BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      036336                                                                                      T9.6:
      036336  104402                                                                                TRAP    C$BSUB
2281 036340  005737  002224'                    TST     EXTFEA                  ;IS EXTENDED FEATURES SOFT. SW SET?
2282 036344  001002                            BNE     4$                      ;BR. IF SOFTWARE SWITCH IS SET (ON)
2283 036346  000137  036566'                    JMP     55$                     ;EXIT SUBTEST
2284 036352  004737  040106'              4$:   JSR     PC,T9REST               ;SET PACKET TO START-UP VALUES
2285
2286 036356                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      036356  012700  000000                                                                        MOV     #PRI00,R0
      036362  104441                                                                                 TRAP    C$SPRI
2287 036364  012703  002762'                    MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
2288 036370  012704  037010'                    MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2289 036374  012764  000012  000006            MOV     #10.,PKBCNT(R4)         ;START WITH EXTENDED FEATURES VALUE
2290 036402                                5$:
2291 036402                                    BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      036402  104404                                                                                TRAP    C$BSEG
2292
2293 036404  004737  015604'                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2294 036410  103405                            BCS     10$                     ;BR IF SOFT INIT = OK
2298 036412  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2299 036414                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      036414  104455                                                                                TRAP    C$ERDF
      036416  001631                                                                                .WORD   921
      036420  003644'                                                                               .WORD   SFIERR
      036422  011644'                                                                               .WORD   SFIMSG
2300 036424  005037  002220'              10$:  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2301 036430  005037  002222'                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2302 036434  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2303 036440  004737  016146'                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2304 036444  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2305 036446  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2309 036450                                    ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      036450  104455                                                                                TRAP    C$ERDF
      036452  001632                                                                                .WORD   922
      036454  037527'                                                                               .WORD   T9SSR
      036456  011656'                                                                               .WORD   PKTSSR
2310 036460  005237  002220'                    INC     FATFLG                  ;SET FATAL ERROR FLAG
2311 036464                                15$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      036464  104406                                                                                TRAP    C$CLP1
2312 036466                                    ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      036466  104410                                                                                TRAP    C$ESCAPE
      036470  000056                                                                                .WORD   10000$-.
2313 036472  005737  002222'                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2314 036476  001004                            BNE     22$                     ;BRANCH IF YES
2318 036500                                    ERRHRD  ERRNO,T9NINT,PKTSSR
      036500  104456                                                                                TRAP    C$ERHRD
      036502  001633                                                                                .WORD   923
```

```
                036504  037616'                                                                 .WORD   T9NINT
                036506  011656'                                                                 .WORD   PKTSSR
2319            036510  016501  000002          22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
2320            036514  012702  000200                  MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
2321            036520  032701  000100                  BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
2322            036524  001402                          BEQ     25$             ;BRANCH IF NOT OFF-LINE
2323            036526  052702  000100                  BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
2324            036532  020201                  25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
2325            036534  001404                          BEQ     30$             ;OKAY IF MATCH
2329            036536                                  ERRHRD  ERRNO,T9NBA,PKTSSR      ;NBA NOT ZERO
                036536  104456                                                          TRAP    C$ERHRD
                036540  001634                                                          .WORD   924
                036542  037066'                                                         .WORD   T9NBA
                036544  011656'                                                         .WORD   PKTSSR
2330            036546                  30$:
2331            036546                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                036546                                                                  10000$:
                036546  104405                                                          TRAP    C$ESEG
2332
2333            036550  012364  000006          MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
2334            036554  020327  003060'         CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
2335            036560  103002                  BHIS    55$                     ;BRANCH IF ALL DATA DONE
2336            036562  000137  036402'         JMP     5$                      ;BRANCH TILL BACK TO ZERO
2337
2338            036566                  55$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                036566                                                                  L10073:
                036566  104403                                                          TRAP    C$ESUB
2339
2340
2341                                    ;+
2342                                    ;
2343                                    ;TEST 9, SUBTEST 7
2344                                    ;
2345                                    ;TEST WRITE CHARACTERISTICS WITH/WITHOUT INTERRUPTS ENABLED
2346                                    ;
2347                                    ;-
2348
2349            036570                          BGNSUB                          ;/////////////  BEGIN SUBTEST  ////////////
                036570                                                                  T9.7:
                036570  104402                                                          TRAP    C$BSUB
2350
2351            036572                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                036572  012700  000000                                                  MOV     #PRI00,R0
                036576  104441                                                          TRAP    C$SPRI
2352            036600  012704  037010'         MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2353            036604  004737  040106'         JSR     PC,T9REST               ;SET UP A VALID PACKET
2354            036610  004737  015604'         JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2355            036614  103405                  BCS     10$                     ;BR IF SOFT INIT = OK
2359            036616  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2360            036620                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                036620  104455                                                          TRAP    C$ERDF
                036622  001635                                                          .WORD   925
                036624  003644'                                                         .WORD   SFIERR
                036626  011644'                                                         .WORD   SFIMSG
2361            036630  005037  002222'         10$:    CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
2362            036634  052714  000200                  BIS     #BIT7,(R4)      ;ENABLE INTERRUPTS
2363            036640  010465  000000                  MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
```

```
2364 036644  004737  016146'              JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
2365 036650  103405                       BCS     15$                 ;BR IF CARRY SET (GOOD RETURN)
2366 036652  010001                       MOV     R0,R1               ;SAVE CONTENTS OF TSSR
2370 036654                               ERRDF   ERRNO,T9SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     036654  104455                                                           TRAP    C$ERDF
     036656  001636                                                           .WORD   926
     036660  037527'                                                          .WORD   T9SSR
     036662  011656'                                                          .WORD   PKTSSR
2371 036664                        15$:   CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
     036664  104406                                                           TRAP    C$CLP1
2372 036666                               ESCAPE  SUB                 ;BY-PASS SUBTEST IF FATAL ERROR
     036666  104410                                                           TRAP    C$ESCAPE
     036670  000102                                                           .WORD   L10074-.
2373 036672  005737  002222'              TST     INTRECV             ;DID AN INTERRUPT OCCUR ?
2374 036676  001004                       BNE     22$                 ;BRANCH IF YES
2378 036700                               ERRHRD  ERRNO,T9NINT,PKTSSR
     036700  104456                                                           TRAP    C$ERHRD
     036702  001637                                                           .WORD   927
     036704  037616'                                                          .WORD   T9NINT
     036706  011656'                                                          .WORD   PKTSSR
2379 036710                        22$:   CKLOOP                      ;LOOP ON ERROR ?
     036710  104406                                                           TRAP    C$CLP1
2380
2381 036712  005037  002222'              CLR     INTRECV             ;CLEAR INTERRUPT RECEIVED FLAG
2382 036716  042714  000200               BIC     #BIT7,(R4)          ;DISABLE INTERRUPTS
2383 036722  010465  000000               MOV     R4,TSDB(R5)         ;SET THE PACKET ADDRESS
2384 036726  004737  016146'              JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
2385 036732  103405                       BCS     25$                 ;BR IF CARRY SET (GOOD RETURN)
2386 036734  010001                       MOV     R0,R1               ;SAVE CONTENTS OF TSSR
2390 036736                               ERRDF   ERRNO,T9SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     036736  104455                                                           TRAP    C$ERDF
     036740  001640                                                           .WORD   928
     036742  037527'                                                          .WORD   T9SSR
     036744  011656'                                                          .WORD   PKTSSR
2391 036746                        25$:   CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
     036746  104406                                                           TRAP    C$CLP1
2392 036750                               ESCAPE  SUB                 ;BY-PASS SUBTEST IF FATAL ERROR
     036750  104410                                                           TRAP    C$ESCAPE
     036752  000020                                                           .WORD   L10074-.
2393 036754  005737  002222'              TST     INTRECV             ;DID AN INTERRUPT OCCUR ?
2394 036760  001404                       BEQ     30$                 ;BRANCH IF NOT
2398 036762                               ERRHRD  ERRNO,T9INT,PKTSSR
     036762  104456                                                           TRAP    C$ERHRD
     036764  001641                                                           .WORD   929
     036766  037707'                                                          .WORD   T9INT
     036770  011656'                                                          .WORD   PKTSSR
2399 036772                        30$:
2400 036772                               ENDSUB                      ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     036772                                                                 L10074:
     036772  104403                                                           TRAP    C$ESUB
2401
2402 036774                               EXIT    TST                 ;ALL DONE THIS TEST
     036774  104432                                                           TRAP    C$EXIT
     036776  001162                                                           .WORD   L10065-.
2403
2404                                   ;+
2405                                   ;LOCAL STORAGE FOR THIS TEST
```

```
                                        ;-
                                        
037000                                          .BLKB   10-<.-TSV2&7>
037010                                  T9PACKET:                       ;COMMAND PACKET FOR TEST
037010  100204                                  .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
037012  037020'                                 .WORD   T9DATA          ;ADDRESS OF CHARACTERISTICS BLOCK
037014  000000                                  .WORD   0
037016  000010                                  .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
                                        
037020                                  T9DATA:                         ;CHARACTERISTICS DATA BLOCK
037020  037032'                                 .WORD   T9BFR           ;ADDRESS OF MESSAGE BUFFER
037022  000000                                  .WORD   0
037024  000016                                  .WORD   14.             ;LENGTH OF MESSAGE BUFFER
037026  000000  000000                         .WORD   0,0
                                        
037032                                  T9BFR:  .BLKW   8.              ;MESSAGE BUFFER
                                        
                                        ;+
                                        ;
                                        ;TEST DATA FOR SUBTEST TWO
                                        ;
                                        ;DATA HAS FORMAT:
                                        ;
                                        ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
                                        ;       2ND WORD        BITS TO SET FOR TEST
                                        ;
                                        ;-
                                        
037052                                  T92DATA:
037052  000000  037140                          .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
037056  000002  000001                          .WORD   2,BIT0
037062  000004  100100                          .WORD   4,BIT6!BIT15
        037066'                         T92DONE=.
                                        
                                        
                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-
                                        
037066      127     122     111  T9NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
037141      127     122     111  T92REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
037240      127     122     111  T93REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
037333      127     122     111  T94REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
037431      127     122     111  T95REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
037527      103     157     156  T9SSR:  .ASCIZ  'Contents of SSR Incorrect After WRITE CHARACTERISTICS'
037616      105     170     160  T9NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
037707      125     156     145  T9INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
037775      111     156     143  T9TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
040061      103     157     155  TST9ID: .ASCIZ  'Completion Interrupt'
                                                 .EVEN
                                        
                                        
                                        ;+
                                        ;
                                        ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
                                        ;
                                        ;-
```

```
                                        T9REST:
                                                SAVREG                          ;SAVE THE REGISTERS
                                                MOV     #T9PACKET,R1            ;START OF THE PACKET
                                                MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
                                                MOV     #T9DATA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
                                                CLR     (R1)+                   ;EXTENDED ADDRESS
                                                MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
                                                MOV     #T9BFR,(R1)+            ;ADDRESS OF MESSAGE BUFFER
                                                CLR     (R1)+
                                                MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
                                                CLR     (R1)+
                                                CLR     (R1)
                                                CLR     T9BFR                   ;CLEAR 1ST LOC IN MESSAGE BUFFER
                                                RTS     PC                      ;RETURN
                                                ENDTST
                                                                                                L10065:
                                                                                                        TRAP    C$ETST

                                                .SBTTL  TEST  10: BASIC PACKET PROTOCOL

                                        ;       THIS TEST VERIFIES BASIC OPERATION OF THE MESSAGE BUFFER RELEASE
                                        ;       COMMAND, THE FUNCTION OF THE ACK BIT IN THE COMMAND HEADER WORD,
                                        ;       AND THE REGISTER MODIFICATION REFUSED (RMR) LOGIC.
                                        ;
                                        ;
                                        ;+
                                        ;TEST 10 SUBTEST 1
                                        ;
                                        ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
                                        ;PROPERLY AND THAT NO INTERRUPT IS GENERATED EVEN
                                        ;IF THE "IE" BIT IS SET IN THE COMMAND PACKET
                                        ;
                                        ;-
                                                BGNTST
                                                                                                T10::
                                                MOV     #TST10ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
                                                JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
                                                MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
                                        T10LOOP:
                                                BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                                                T10.1:
                                                                                                        TRAP    C$BSUB
                                                JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
                                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                                                        MOV     #PRI00,R0
                                                                                                        TRAP    C$SPRI
                                                MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
                                                MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
                                        5$:
                                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
                                                                                                        TRAP    C$BSEG
                                                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
                                                BCS     10$                     ;BR IF SOFT INIT = OK
```

```
2521 040236  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2522 040240                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     040240  104455                                                                 TRAP    C$ERRDF
     040242  001751                                                                 .WORD   1001
     040244  003644'                                                                .WORD   SFIERR
     040246  011644'                                                                .WORD   SFIMSG
2523 040250  005037  002220'       10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2524 040254  005077  002222'               CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2525 040260  010465  000000               MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2526 040264  004737  016146'               JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2527 040270  103407                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2528 040272  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2532 040274                                 ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     040274  104455                                                                 TRAP    C$ERRDF
     040276  001752                                                                 .WORD   1002
     040300  042650'                                                                .WORD   T10SSR
     040302  011656'                                                                .WORD   PKTSSR
2533 040304  005237  002220'               INC     FATFLG                  ;SET FATAL ERROR FLAG
2534 040310                        15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040310  104406                                                                 TRAP    C$CLP1
2535 040312                                 ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     040312  104410                                                                 TRAP    C$ESCAPE
     040314  000056                                                                 .WORD   10000$-.
2536 040316  005737  002222'               TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2537 040322  001004                        BNE     22$                     ;BRANCH IF YES
2541 040324                                 ERRHRD  ERRNO,T10NINT,PKTSSR
     040324  104456                                                                 TRAP    C$ERHRD
     040326  001753                                                                 .WORD   1003
     040330  042737'                                                                .WORD   T10NINT
     040332  011656'                                                                .WORD   PKTSSR
2542 040334  016501  000002       22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2543 040340  012702  000200               MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2544 040344  032701  000100               BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2545 040350  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2546 040352  052702  000100               BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2547 040356  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2548 040360  001404                        BEQ     30$                     ;OKAY IF MATCH
2552 040362                                 ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
     040362  104456                                                                 TRAP    C$ERHRD
     040364  001754                                                                 .WORD   1004
     040366  042511'                                                                .WORD   T10NBA
     040370  011656'                                                                .WORD   PKTSSR
2553 040372                        30$:
2554 040372                                 ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     040372                                                                         10000$:
     040372  104405                                                                 TRAP    C$ESEG
2555 040374                                 BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     040374  104404                                                                 TRAP    C$BSEG
2556
2557 040376  005037  002222'               CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2558 040402  012737  025252  042332'       MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
2559 040410  012714  100212               MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
2560 040414  010465  000000               MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2561 040420  004737  016146'               JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2562 040424  103407                        BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
2563 040426  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2567 040430                                 ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
```

```
                040430  104455                                                          TRAP    C$ERDF
                040432  001755                                                          .WORD   1005
                040434  042650'                                                        .WORD   T10SSR
                040436  011656'                                                        .WORD   PKTSSR
     2568       040440  005237  002220'         INC     FATFLG                  ;SET FATAL ERROR FLAG
     2569       040444                  45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                040444  104406                                                          TRAP    C$CLP1
     2570       040446  005737  002222'         TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
     2571       040452  001404                 BEQ     52$                     ;BRANCH IF NO
     2575       040454                          ERRHRD  ERRNO,T10INT,PKTSSR
                040454  104456                                                          TRAP    C$ERHRD
                040456  001756                                                          .WORD   1006
                040460  043030'                                                        .WORD   T10INT
                040462  011656'                                                        .WORD   PKTSSR
     2576       040464  016501  000002  52$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
     2577       040470  012702  000200         MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
     2578       040474  032701  000100         BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
     2579       040500  001402                 BEQ     55$                     ;BRANCH IF NOT OFF-LINE
     2580       040502  052702  000100         BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
     2581       040506  020201          55$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
     2582       040510  001404                 BEQ     60$                     ;OKAY IF MATCH
     2586       040512                          ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA  SET
                040512  104456                                                          TRAP    C$ERHRD
                040514  001757                                                          .WORD   1007
                040516  042573'                                                        .WORD   T10NNBA
                040520  011656'                                                        .WORD   PKTSSR
     2587       040522                  60$:
     2588       040522  013701  042332'         MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
     2589       040526  012702  025252         MOV     #025252,R2              ;SET UP EXPECTED DATA
     2590       040532  020102                 CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
     2591       040534  001404                 BEQ     70$                     ;BR. IF OK (EQUAL)
     2595       040536                          ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
                040536  104456                                                          TRAP    C$ERHRD
                040540  001760                                                          .WORD   1008
                040542  042414'                                                        .WORD   T10MBF
                040544  015304'                                                        .WORD   EXPREC
     2596       040546                  70$:
     2597       040546  005737  002220'         TST     FATFLG                  ;ANY FATAL ERRORS
     2598       040552  001403                 BEQ     80$                     ;BR. IF NO FATAL ERRORS
     2599       040554  004737  017014'         JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
     2600       040560                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                040560                                                                  10001$:
                040560  104405                                                          TRAP    C$ESEG
     2601       040562                  80$:
     2602       040562                          ENDSUB                          ;/////////// END SUBTEST ///////////
                040562                                                                  L10076:
                040562  104403                                                          TRAP    C$ESUB
     2603
     2604                                   ;+
     2605                                   ;TEST 10 SUBTEST 2
     2606                                   ;
     2607                                   ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
     2608                                   ;PROPERLY AND THAT THERE IS AN INTERRUPT IF THE "IE"
     2609                                   ;BIT IS SET IN THE COMMAND PACKET AND THE "ERI" BIT
     2610                                   ;IS SET IN THE CHARACTERISTICS DATA PACKET
     2611                                   ;
     2612                                   ;-
```

```
2613 040564                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     040564                                                                                    T10.2:
     040564  104402                                                                           TRAP    C$BSUB
2614
2615 040566  004737  043146'                   JSR     PC,T1ORST               ;SET PACKET TO INITIAL VALUES
2616 040572                                    SETPRI  @PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     040572  012700  000000                                                                   MOV     @PRI00,R0
     040576  104441                                                                           TRAP    C$SPRI
2617 040600  012704  042310'                   MOV     #T1OPACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2618 040604  012764  000010  000006           MOV     @8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
2619 040612                            5$:
2620 040612                                    BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     040612  104404                                                                           TRAP    C$BSEG
2621
2622 040614  004737  015604'                   JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2623 040620  103405                           BCS     10$                     ;BR IF SOFT INIT = OK
2627 040622  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2628 040624                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     040624  104455                                                                           TRAP    C$ERDF
     040626  001761                                                                           .WORD   1009
     040630  003644'                                                                          .WORD   SFIERR
     040632  011644'                                                                          .WORD   SFIMSG
2629 040634  005037  002220'           10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2630 040640  005037  002222'                   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2631 040644  012737  000020  042326'           MOV     #000020,T10DATA+6       ;SET ERI IN CHARACTERISTICS DATA
2632 040652  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2633 040656  004737  016146'                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2634 040662  103407                           BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2635 040664  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2639 040666                                    ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     040666  104455                                                                           TRAP    C$ERDF
     040670  001762                                                                           .WORD   1010
     040672  042650'                                                                          .WORD   T10SSR
     040674  011656'                                                                          .WORD   PKTSSR
2640 040676  005237  002220'                   INC     FATFLG                  ;SET FATAL ERROR FLAG
2641 040702                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040702  104406                                                                           TRAP    C$CLP1
2642 040704                                    ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     040704  104410                                                                           TRAP    C$ESCAPE
     040706  000056                                                                           .WORD   10000$-.
2643 040710  005737  002222'                   TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2644 040714  001004                           BNE     22$                     ;BRANCH IF YES
2648 040716                                    ERRHRD  ERRNO,T10NINT,PKTSSR
     040716  104456                                                                           TRAP    C$ERHRD
     040720  001763                                                                           .WORD   1011
     040722  042737'                                                                          .WORD   T10NINT
     040724  011656'                                                                          .WORD   PKTSSR
2649 040726  016501  000002           22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2650 040732  012702  000200                   MOV     @SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2651 040736  032701  000100                   BIT     @OFL,R1                 ;IS OFF-LINE BIT SET ?
2652 040742  001402                           BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2653 040744  052702  000100                   BIS     @OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2654 040750  020201                    25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2655 040752  001404                           BEQ     30$                     ;OKAY IF MATCH
2659 040754                                    ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
     040754  104456                                                                           TRAP    C$ERHRD
     040756  001764                                                                           .WORD   1012
```

```
2707 041146  004737  017014'                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2708 041152                          80$:
2709 041152                                  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     041152                                                                      10001$:
     041152  104405                                                                TRAP    C$ESEG
2710 041154                                  ENDSUB                          ;////////////// END SUBTEST //////////////
     041154                                                                      L10077:
     041154  104403                                                                TRAP    C$ESUB
2711
2712                                 ;+
2713                                 ;TEST 10 SUBTEST 3
2714                                 ;
2715                                 ;CHECKS THAT THE CPU GIVES UP OWNERSHIP OF THE MESSAGE BUFFER
2716                                 ;AFTER THE MESSAGE BUFFER RELEASE, AND THAT FOLLOWING COMMANDS
2717                                 ;WORK CORRECTLY
2718                                 ;
2719                                 ;-
2720 041156                                  BGNSUB                          ;////////////// BECIN SUBTEST //////////////
     041156                                                                      T10.3:
     041156  104402                                                                TRAP    C$BSUB
2721
2722 041160  004737  043146'                 JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
2723 041164                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     041164  012700  000000                                                         MOV     #PRI00,R0
     041170  104441                                                                 TRAP    C$SPRI
2724 041172  012704  042310'                 MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2725 041176  012764  000010  000006         MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
2726 041204                          5$:
2727 041204                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     041204  104404                                                                TRAP    C$BSEG
2728
2729 041206  004737  015604'                 JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2730 041212  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
2731 041214  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2735 041216                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     041216  104455                                                                 TRAP    C$ERDF
     041220  001771                                                                 .WORD   1017
     041222  003644'                                                                .WORD   SFIERR
     041224  011644'                                                                .WORD   SFIMSG
2736 041226  005037  002220'         10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2737 041232  005037  002222'                 CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2738 041236  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2739 041242  004737  016146'                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2740 041246  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2741 041250  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2745 041252                                  ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     041252  104455                                                                 TRAP    C$ERDF
     041254  001772                                                                 .WORD   1018
     041256  042650'                                                                .WORD   T10SSR
     041260  011656'                                                                .WORD   PKTSSR
2746 041262  005237  002220'                 INC     FATFLG                  ;SET FATAL ERROR FLAG
2747 041266                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041266  104406                                                                 TRAP    C$CLP1
2748 041270                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     041270  104410                                                                 TRAP    C$ESCAPE
     041272  000056                                                                 .WORD   10000$-.
2749 041274  005737  002222'                 TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
```

```
2750 041300  001004                    BNE     22$                 ;BRANCH IF YES
2754 041302                            ERRHRD  ERRNO,T10NINT,PKTSSR
     041302  104456                                                        TRAP    C$ERHRD
     041304  001773                                                        .WORD   1019
     041306  042737'                                                      .WORD   T10NINT
     041310  011656'                                                      .WORD   PKTSSR
2755 041312  016501  000002      22$:  MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
2756 041316  012702  000200            MOV     #SSR,R2             ;EXPECTED CONTENTS OF TSSR
2757 041322  032701  000100            BIT     #OFL,R1             ;IS OFF-LINE BIT SET ?
2758 041326  001402                    BEQ     25$                 ;BRANCH IF NOT OFF-LINE
2759 041330  052702  000100            BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
2760 041334  020201            25$:    CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
2761 041336  001404                    BEQ     30$                 ;OKAY IF MATCH
2765 041340                            ERRHRD  ERRNO,T10NBA,PKTSSR ;NBA NOT ZERO
     041340  104456                                                        TRAP    C$ERHRD
     041342  001774                                                        .WORD   1020
     041344  042511'                                                      .WORD   T10NBA
     041346  011656'                                                      .WORD   PKTSSR
2766 041350                    30$:
2767 041350                            ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     041350                                                                 10000$:
     041350  104405                                                        TRAP    C$ESEG
2768 041352                            BGNSEG                      ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     041352  104404                                                        TRAP    C$BSEG
2769 041354  004737  043146'           JSR     PC,T10RST           ;RESET PACKETS AND COMMANDS
2770 041360  005037  002222'           CLR     INTRECV             ;CLEAR INTERRUPT RECEIVED FLAG
2771 041364  012737  025252  042332'  MOV     #025252,T10BFR      ;WIPE OUT MESSAGE BUFFER AREA
2772 041372  012714  100212            MOV     #100212,(R4)        ;SET COMMAND PACKET TO MESS BUF REL
2773 041376  010465  000000            MOV     R4,TSDB(R5)         ;SET THE PACKET ADDRESS
2774 041402  004737  016146'           JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
2775 041406  103407                    BCS     45$                 ;BR IF CARRY SET (GOOD RETURN)
2776 041410  010001                    MOV     R0,R1               ;SAVE CONTENTS OF TSSR
2780 041412                            ERRDF   ERRNO,T10SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
     041412  104455                                                        TRAP    C$ERDF
     041414  001775                                                        .WORD   1021
     041416  042650'                                                      .WORD   T10SSR
     041420  011656'                                                      .WORD   PKTSSR
2781 041422  005237  002220'           INC     FATFLG              ;SET FATAL ERROR FLAG
2782 041426                    45$:    CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
     041426  104406                                                        TRAP    C$CLP1
2783 041430  005737  002222'           TST     INTRECV             ;DID AN INTERRUPT OCCUR ?
2784 041434  001404                    BEQ     52$                 ;BRANCH IF NO
2788 041436                            ERRHRD  ERRNO,T10INT,PKTSSR
     041436  104456                                                        TRAP    C$ERHRD
     041440  001776                                                        .WORD   1022
     041442  043030'                                                      .WORD   T10INT
     041444  011656'                                                      .WORD   PKTSSR
2789 041446  016501  000002      52$:  MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
2790 041452  012702  000200            MOV     #SSR,R2             ;EXPECTED CONTENTS OF TSSR
2791 041456  032701  000100            BIT     #OFL,R1             ;IS OFF-LINE BIT SET ?
2792 041462  001402                    BEQ     55$                 ;BRANCH IF NOT OFF-LINE
2793 041464  052702  000100            BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
2794 041470  020201            55$:    CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
2795 041472  001404                    BEQ     60$                 ;OKAY IF MATCH
2799 041474                            ERRHRD  ERRNO,T10NNBA,PKTSSR ;NBA NOT SET
     041474  104456                                                        TRAP    C$ERHRD
     041476  001777                                                        .WORD   1023
```

```
        041500  042573'                                                         .WORD   T10NNBA
        041502  011656'                                                         .WORD   PKTSSR
  2800  041504                          10$:
  2801  041504  013701  042332'                MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
  2802  041510  012702  025252                 MOV     #025252,R2              ;SET UP EXPECTED DATA
  2803  041514  020102                         CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
  2804  041516  001404                         BEQ     70$                     ;BR, IF OK (EQUAL)
  2808  041520                                  ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
        041520  104456                                                         TRAP    C$ERHRD
        041522  002000                                                         .WORD   1024
        041524  042414'                                                        .WORD   T10MBF
        041526  015304'                                                        .WORD   EXPREC
  2809
  2810  041530                          70$:    CKLOOP                          ;LOOP ON ERROR IF FLAG SET
        041530  104406                                                         TRAP    C$CLP1
  2811
  2812  041532  005037  002222'                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
  2813  041536  004737  043146'                JSR     PC,T10RST               ;RESET THE PACKETS AND COMMANDS
  2814  041542  042714  100000                 BIC     #100000,(R4)            ;CLEAR THE ACK BIT
  2815  041546  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
  2816  041552  004737  016146'                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  2817  041556  103407                         BCS     75$                     ;BR IF CARRY SET (GOOD RETURN)
  2818  041560  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  2822  041562                                  ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        041562  104455                                                         TRAP    C$ERDF
        041564  002001                                                         .WORD   1025
        041566  042650'                                                        .WORD   T10SSR
        041570  011656'                                                        .WORD   PKTSSR
  2823  041572  005237  002220'                INC     FATFLG                  ;SET FATAL ERROR FLAG
  2824  041576                          75$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        041576  104406                                                         TRAP    C$CLP1
  2825  041600                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
        041600  104410                                                         TRAP    C$ESCAPE
        041602  000062                                                         .WORD   10001$-.
  2826  041604  005737  002222'                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
  2827  041610  001006                         BNE     82$                     ;BRANCH IF YES
  2831  041612  016500  000002                 MOV     TSSR(R5),R0             ;GET TSSR FOR ERROR REPORT
  2832  041616                                  ERRHRD  ERRNO,T10NINT,PKTSSR
        041616  104456                                                         TRAP    C$ERHRD
        041620  002002                                                         .WORD   1026
        041622  042737'                                                        .WORD   T10NINT
        041624  011656'                                                        .WORD   PKTSSR
  2833  041626  016501  000002          82$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
  2834  041632  012702  000200                 MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
  2835  041636  032701  000100                 BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
  2836  041642  001402                         BEQ     85$                     ;BRANCH IF NOT OFF-LINE
  2837  041644  052702  000100                 BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
  2838  041650  020201          85$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
  2839  041652  001404                         BEQ     90$                     ;OKAY IF MATCH
  2843  041654                                  ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
        041654  104456                                                         TRAP    C$ERHRD
        041656  002003                                                         .WORD   1027
        041660  042650'                                                        .WORD   T10SSR
        041662  011656'                                                        .WORD   PKTSSR
  2844  041664                          90$:
  2845  041664                          ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<
        041664                                                                  10001$:
```

```
                041664  104405                                                  TRAP    C$ESEG
2846            041666  005737  002220'         TST     FATFLG                  ;ANY FATAL EPRORS
2847            041672  001403                  BEQ     95$                     ;BR, IF NO FATAL ERRORS
2848            041674  004737  017014'         JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2849
2850            041700                          BGNSEG                          ;<<<<<<<<<<<<< BGN SEGMENT <<<<<<<<<<<<<
                041700  104404                                                  TRAP    C$BSEG
2851            041702  005037  002222'  95$:   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2852            041706  004737  043146'         JSR     PC,T10RST               ;RESET THE PACKETS AND COMMANDS
2853            041712  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2854            041716  004737  016146'         JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2855            041722  103407                  BCS     100$                    ;BR IF CARRY SET (GOOD RETURN)
2856            041724  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2860            041726                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                041726  104455                                                  TRAP    C$ERDF
                041730  002004                                                  .WORD   1028
                041732  042650'                                                 .WORD   T10SSR
                041734  011656'                                                 .WORD   PKTSSR
2861            041736  005237  002220'         INC     FATFLG                  ;SET FATAL ERROR FLAG
2862            041742                  100$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                041742  104406                                                  TRAP    C$CLP:
2863            041744                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
                041744  104410                                                  TRAP    C$ESCAPE
                041746  000062                                                  .WORD   10002$-.
2864            041750  005737  002222'         TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2865            041754  001006                  BNE     112$                    ;BRANCH IF YES
2869            041756  016500  000002          MOV     TSSR(R5),R0             ;GET TSSR FOR ERROR REPORT
2870            041762                          ERRHRD  ERRNO,T10NINT,PKTSSR
                041762  104456                                                  TRAP    C$ERHRD
                041764  002005                                                  .WORD   1029
                041766  042737'                                                 .WORD   T10NINT
                041770  011656'                                                 .WORD   PKTSSR
2871            041772  016501  000002  112$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2872            041776  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2873            042002  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2874            042006  001402                  BEQ     125$                    ;BRANCH IF NOT OFF-LINE
2875            042010  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2876            042014  020201          125$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2877            042016  001405                  BEQ     130$                    ;OKAY IF MATCH
2881            042020                          ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
                042020  104456                                                  TRAP    C$ERHRD
                042022  002006                                                  .WORD   1030
                042024  042650'                                                 .WORD   T10SSR
                042026  011656'                                                 .WORD   PKTSSR
2882            042030                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                042030                                                  10002$:
                042030  104405                                                  TRAP    C$ESEG
2883            042032  005737  002220'  130$:   TST     FATFLG                  ;CHECK FATAL FLAG FOR SET
2884            042036  001402                  BEQ     140$                    ;BR, IF NOT SET (OK)
2885            042040  004737  017014'         JSR     PC,CKDROP               ;DROP DEVICE IF INDICATED
2886            042044                  140$:
2887            042044                          ENDSUB                          ;////////////// END SUBTEST //////////// 
                042044                                                  L10100:
                042044  104403                                                  TRAP    C$ESUB
2888            042046          30$:
2889
2890                            ;*
```

```
                                        ;TEST 10 SUBTEST 4
                                        ;
                                        ;CHECKS THAT THE REGISTER MODIFICATION REFUSED (RMR) BIT IN
                                        ;THE TSSR WILL BE SET IF A WRITE CHARACTERISTICS COMMAND
                                        ;BEING EXECUTED AND ANOTHER "WC" COMMAND IS ATTEMPTED
                                        ;
                                        ;-

042046                                          BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
042046                                                                                  110.4:
042046  104402                                                                            TRAP    C$BSUB

042050  004737  043220'                        JSR     PC,T10RT2               ;SET SECOND PACKET UP
042054  004737  043146'                        JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
042060                                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
042060  012700  000000                                                                    MOV     #PRI00,R0
042064  104441                                                                            TRAP    C$SPRI
042066  012704  042310'                        MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
042072  012703  042352'                        MOV     #T10PKT,R3              ;GET THE ADDRESS OF 2ND CMD PACKET
042076  012764  000010  000006                MOV     #8.,PKSCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
042104  012763  000010  000006                MOV     #8.,PKBCNT(R3)          ;START WITH MINIMUM ALLOWABLE VALUE
042112                                  5$:
042112                                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
042112  104404                                                                          TRAP    C$BSEG
042114  004737  015604'                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
042120  103405                                  BCS     10$                     ;BR IF SOFT INIT = OK
042122  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
042124                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
042124  104455                                                                          TRAP    C$ERDF
042126  002007                                                                          .WORD   1031
042130  003644'                                                                         .WORD   SFIERR
042132  011644'                                                                         .WORD   SFIMSG
042134  005037  002220'                10$:     CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
042140  005037  002222'                        CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
042144  010465  000000                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
042150  010365  000000                        MOV     R3,TSDB(R5)             ;SECOND COMMAND PACKET
042154  004737  016060'                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
042160  016501  000002                        MOV     T3SR(R5),R1             ;GET CONTENTS OF TSSR REGISTER
042164  032701  000200                        BIT     #SSR,R1                 ;CHECK FOR SSR (TSSR) SET
042170  001006                                BNE     15$                     ;BR, IF SSR SET (GOOD)
042172                                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
042172  104455                                                                          TRAP    C$ERDF
042174  002010                                                                          .WORD   1032
042176  042650'                                                                         .WORD   T10SSR
042200  011656'                                                                         .WORD   PKTSSR
042202  005237  002220'                        INC     FATFLG                  ;SET FATAL ERROR FLAG
042206                                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
042206  104406                                                                          TRAP    C$CLP1
042210                                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
042210  104410                                                                          TRAP    C$ESCAPE
042212  000056                                                                          .WORD   10000$-.
042214  005737  002222'                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
042220  001004                                  BNE     22$                     ;BRANCH IF YES

042222                                          ERRHRD  ERRNO,T10N1NT,PKTSSR
042222  104456                                                                          TRAP    C$ERHRD
```

```
        042224  002011                                                          .WORD   1033
        042226  042737'                                                        .WORD   T10NINT
        042230  011656'                                                        .WORD   PKTSSR
   2940 042232  016501  000002          22$:    MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
   2941 042236  012702  110200                  MOV     #SSR!RMR!SC,R2      ;EXPECTED CONTENTS OF TSSR
   2942 042242  032701  000100                  BIT     #OFL,R1             ;IS OFF-LINE BIT SET ?
   2943 042246  001402                          BEQ     25$                 ;BRANCH IF NOT OFF-LINE
   2944 042250  052702  000100                  BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
   2945 042254  020201                  25$:    CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
   2946 042256  001404                          BEQ     30$                 ;OKAY IF MATCH
   2950 042260                                  ERRHRD  ERRNO,T10SSR,PKTSSR ;NBA NOT ZERO
        042260  104456                                                          TRAP    C$ERHRD
        042262  002012                                                          .WORD   1034
        042264  042650'                                                        .WORD   T10SSR
        042266  011656'                                                        .WORD   PKTSSR
   2951 042270                          30$:
   2952 042270                                  ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        042270                                                                  10000$:
        042270  104405                                                          TRAP    C$ESEG
   2953 042272                                  ENDSUB                      ;////////////// END SUBTEST //////////////
        042272                                                                  L10101:
        042272  104403                                                          TRAP    C$ESUB
   2954 042274                                  EXIT    TST                 ;ALL DONE WITH THIS TEST
        042274  104432                                                          TRAP    C$EXIT
        042276  000774                                                          .WORD   L10075-.
   2955
   2956                                 ;+
   2957                                 ;LOCAL STORAGE FOR THIS TEST
   2958                                 ;-
   2959
   2961 042300                                  .BLKB   10-<.-TSV2&7>
   2963 042310                          T10PACKET:                          ;COMMAND PACKET FOR TEST
   2964 042310  100204                          .WORD   100204              ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
   2965 042312  042320'                        .WORD   T10DATA             ;ADDRESS OF CHARACTERISTICS BLOCK
   2966 042314  000000                          .WORD   0
   2967 042316  000010                          .WORD   8.                  ;STARTING VALUE OF BLOCK SIZE
   2968
   2969 042320                          T10DATA:                            ;CHARACTERISTICS DATA BLOCK
   2970 042320  042332'                         .WORD   T10BFR              ;ADDRESS OF MESSAGE BUFFER
   2971 042322  000000                          .WORD   0
   2972 042324  000016                          .WORD   14.                 ;LENGTH OF MESSAGE BUFFER
   2973 042326  000000  000000                  .WORD   0,0
   2974
   2975 042332                          T10BFR: .BLKW   8.                  ;MESSAGE BUFFER
   2976
   2977                                 ;+
   2978                                 ;
   2979                                 ;TEST DATA FOR SUBTEST FOUR
   2980                                 ;
   2981 042352                          T10PKT:                             ;COMMAND PACKET FOR TEST
   2982 042352  100204                          .WORD   100204              ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
   2983 042354  042362'                         .WORD   T10DTA              ;ADDRESS OF CHARACTERISTICS BLOCK
   2984 042356  000000                          .WORD   0
   2985 042360  000010                          .WORD   8.                  ;STARTING VALUE OF BLOCK SIZE
   2986
   2987 042362                          T10DTA:                             ;CHARACTERISTICS DATA BLOCK
   2988 042362  042374'                         .WORD   T10BUFR             ;ADDRESS OF MESSAGE BUFFER
```

```
2989 042364  000000                          .WORD   0
2990 042366  000016                          .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
2991 042370  000000  000000                  .WORD   0,0
2992
2993 042374                          T10BUFR:.BLKW   8.                      ;MESSAGE BUFFER
2994
2995                                 ;*
2996                                 ;LOCAL TEXT MESSAGES FOR TEST
2997                                 ;-
2998
2999
3000 042414     115     145     163 T10MBF: .ASCIZ  'Message Buffer Modified after MESSAGE BUFFER RELEASE Command'
3001 042511     116     102     101 T10NBA: .ASCIZ  'NBA Not Clear After WRITE CHARACTERISTICS Command'
3002 042573     116     102     101 T10NNBA:        .ASCIZ  'NBA Set After MESSAGE BUFFER RELEASE Command'
3003 042650     103     157     156 T10SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
3004 042737     105     170     160 T10NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
3005 043030     125     156     145 T101NT: .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
3006 043117     102     141     163 TST10ID:        .ASCIZ  'Basic Packet Protocol'
3007                                 .EVEN
3008
3009
3010
3011                                 ;*
3012                                 ;
3013                                 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3014                                 ;
3015                                 ;-
3016
3017 043146                          T10RST:
3018 043146                                 SAVREG                          ;SAVE THE REGISTERS
3019 043152  012701  042310'                MOV     #T10PACKET,R1           ;START OF THE PACKET
3020 043156  012721  100204                 MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
3021 043162  012721  042320'                MOV     #T10DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
3022 043166  005021                         CLR     (R1)+                   ;EXTENDED ADDRESS
3023 043170  012721  000010                 MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
3024 043174  012721  042332'                MOV     #T10BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
3025 043200  005021                         CLR     (R1)+
3026 043202  012721  000016                 MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
3027 043206  005021                         CLR     (R1)+
3028 043210  005011                         CLR     (R1)
3029 043212  005037  042332'                CLR     T10BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
3030 043216  000207                         RTS     PC                      ;RETURN
3031                                 ;*
3032                                 ;
3033                                 ;ROUTINE TO RESTORE COMMAND PACKET #2 TO START-UP (DEFAULT) VALUES
3034                                 ;
3035                                 ;-
3036
3037 043220                          T10RT2:
3038 043220                                 SAVREG                          ;SAVE THE REGISTERS
3039 043224  012701  042352'                MOV     #T10PKT,R1              ;START OF THE PACKET
3040 043230  012721  100204                 MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
3041 043234  012721  042362'                MOV     #T10DTA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
3042 043240  005021                         CLR     (R1)+                   ;EXTENDED ADDRESS
3043 043242  012721  000010                 MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
3044 043246  012721  042374'                MOV     #T10BUFR,(R1)+          ;ADDRESS OF MESSAGE BUFFER
3045 043252  005021                         CLR     (R1)+
```

```
3046 043254  012721  000016                  MOV     #14.,(R1)+      ;LENGTH OF MESSAGE BUFFER
3047 043260  005021                          CLR     (R1)+
3048 043262  005011                          CLR     (R1)
3049 043264  005037  042374'                CLR     T10BUFR         ;CLEAR 1ST LOC IN MESSAGE BUFFER
3050 043270  000207                          RTS     PC              ;RETURN
3051 043272                                  ENDTST
     043272                                                                  L10075:
     043272  104401                                                                  TRAP    C$ETST
3052
3053                                         .SBTTL  TEST  11: NON-TAPE MOTION COMMANDS
3054
3055                                 ;+
3056                                 ;
3057                                 ;THIS TEST VERIFIES PROPER OPERATION OF THE INITIALIZE
3058                                 ;COMMAND. TWO SUBTESTS ARE USED. THE FIRST VERIFIES THAT
3059                                 ;THE COMMAND RUNS TO COMPLETION AND STORES A VALID
3060                                 ;MESSAGE PACKET. THE SECOND VERIFIES THAT NON-ZERO
3061                                 ;VALUES IN THE COMMAND MODE FIELD CAUSES COMMAND REJECT.
3062                                 ;
3063                                 ;-
3064
3065 043274                                  BGNTST
     043274                                                                  T11::
3070 043274  012700  045302'                 MOV     #TST11ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
3071 043300  004737  016322'                 JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
3072 043304  012737  000024  002214'         MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
3073 043312                          T11LOOP:
3074 043312                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
     043312                                                                          T11.1:
     043312  104402                                                                  TRAP    C$BSUB
3075
3076 043314                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     043314  012700  000000                                                          MOV     #PRI00,R0
     043320  104441                                                                  TRAP    C$SPRI
3077 043322  004737  015604'                 JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
3078 043326  103405                          BCS     3$                      ;BR IF SOFT INIT = OK
3082 043330  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3083 043332                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     043332  104455                                                                  TRAP    C$ERDF
     043334  002115                                                                  .WORD   1101
     043336  003644'                                                                 .WORD   SFIERR
     043340  011644'                                                                 .WORD   SFIMSG
3084 043342                          3$:
3085 043342  012704  044530'                 MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
3086 043346  004737  010472'                 JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3087 043352  103404                          BCS     4$                      ;BR, IF COMMAND ISSUED OK
3091 043354                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     043354  104456                                                                  TRAP    C$ERHRD
     043356  002116                                                                  .WORD   1102
     043360  005050'                                                                 .WORD   WRTMSG
     043362  011644'                                                                 .WORD   SFIMSG
3092 043364                          4$:
3093 043364  004737  045334'                 JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3094 043370  012704  044460'                 MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3095 043374                          5$:
3096 043374                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     043374  104404                                                                  TRAP    C$BSEG
```

```
3097
3098 043376  005037  002220'        10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
3099 043402  005037  002222'                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3100 043406  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
3101 043412  004737  016146'                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3102 043416  103407                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3103 043420  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3107 043422                                ERRDF   ERRNO,T11SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     043422  104455                                                                TRAP    C$ERDF
     043424  002117                                                                .WORD   1103
     043426  045024'                                                               .WORD   T11SSR
     043430  011656'                                                               .WORD   PKTSSR
3108 043432  005237  002220'                INC     FATFLG                  ;SET FATAL ERROR FLAG
3109 043436                        15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043436  104406                                                                TRAP    C$CLP1
3110 043440                                ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     043440  104410                                                                TRAP    C$ESCAPE
     043442  000074                                                                .WORD   10000$-.
3111 043444  005737  002222'                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
3112 043450  001004                        BNE     22$                     ;BRANCH IF YES
3116 043452                                ERRHRD  ERRNO,T11NINT,PKTSSR
     043452  104456                                                                TRAP    C$ERHRD
     043454  002120                                                                .WORD   1104
     043456  045154'                                                               .WORD   T11NINT
     043460  011656'                                                               .WORD   PKTSSR
3117 043462  016501  000002        22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
3118 043466  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
3119 043472  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
3120 043476  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
3121 043500  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
3122 043504  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
3123 043506  001404                        BEQ     30$                     ;OKAY IF MATCH
3127 043510                                ERRHRD  ERRNO,T11NBA,PKTSSR     ;NBA NOT ZERO
     043510  104456                                                                TRAP    C$ERHRD
     043512  002121                                                                .WORD   1105
     043514  044572'                                                               .WORD   T11NBA
     043516  011656'                                                               .WORD   PKTSSR
3128 043520                        30$:
3129 043520  004737  010724'        35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
3130 043524  103405                        BCS     59$                     ;RAM OK GO ON
3134 043526                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     043526  104456                                                                TRAP    C$ERHRD
     043530  002122                                                                .WORD   1106
     043532  004737'                                                               .WORD   PKTRAM
     043534  015320'                                                               .WORD   RAMERR
3135 043536                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     043536                                                                    10000$:
     043536  104405                                                                TRAP    C$ESEG
3136
3137
3138 043540                        59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     043540                                                                    L10103:
     043540  104403                                                                TRAP    C$ESUB
3139
3140 043542  005737  002220'                TST     FATFLG                  ;ANY FATAL ERRORS ?
3141 043546  001402                        BEQ     60$                     ;BRANCH IF NOT
3142 043550  004737  017014'                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
```

```
3143 043554                          60$:
3144
3145                                 ;+
3146                                 ;
3147                                 ;TEST 11, SUBTEST 2
3148                                 ;
3149                                 ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
3150                                 ;INITIALIZE COMMAND TO BE REJECTED
3151                                 ;
3152                                 ;-
3153
3154 043554                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      043554                                                                        T11.2:
      043554  104402                                                                 TRAP    C$BSUB
3155
3156 043556                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      043556  012700  000000                                                        MOV     #PRI00,R0
      043562  104441                                                                TRAP    C$SPRI
3157 043564                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      043564  104404                                                                TRAP    C$BSEG
3158
3159
3160 043566  004737  015604'                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
3161 043572  103405                             BCS     3$                      ;BR IF SOFT INIT = OK
3165 043574  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3166 043576                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      043576  104455                                                                TRAP    C$ERDF
      043600  002123                                                                .WORD   1107
      043602  003644'                                                               .WORD   SFIERR
      043604  011644'                                                               .WORD   SFIMSG
3167 043606                          3$:
3168 043606  012704  044530'                    MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
3169 043612  004737  010472'                    JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3170 043616  103404                             BCS     4$                      ;BR, IF COMMAND ISSUED OK
3174 043620                                     ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      043620  104456                                                                TRAP    C$ERHRD
      043622  002124                                                                .WORD   1108
      043624  005050'                                                               .WORD   WRTMSG
      043626  011644'                                                               .WORD   SFIMSG
3175 043630                          4$:
3176 043630  004737  045334'                    JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3177 043634  012704  044460'                    MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3178 043640                          5$:
3179 043640  005037  002222'          10$:      CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3180 043644  052714  007400                     BIS     #P.MODE,(R4)            ;NON-ZERO COMMAND MODE BITS
3181 043650  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
3182 043654  004737  016146'                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3183 043660  103405                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3184 043662  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3188 043664                                     ERRDF   ERRNO,T11SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      043664  104455                                                                TRAP    C$ERDF
      043666  002125                                                                .WORD   1109
      043670  045024'                                                               .WORD   T11SSR
      043672  011656'                                                               .WORD   PKTSSR
3189 043674                          15$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      043674  104406                                                                TRAP    C$CLP1
3190 043676                                     ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
```

```
                043676  104410                                                          TRAP    C$ESCAPE
                043700  000074                                                          .WORD   10000$-.
3191            043702  005737  002322'         TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
3192            043706  001004                  BNE     22$             ;BRANCH IF YES
3196            043710                          ERRHRD  ERRNO,T11NINT,PKTSSR
                043710  104456                                                          TRAP    C$ERHRD
                043712  002126                                                          .WORD   1110
                043714  045154'                                                         .WORD   T11NINT
                043716  011656'                                                         .WORD   PKTSSR
3197            043720  016501  000002  22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
3198            043724  012702  100206          MOV     #SC!SSR!TSREJ,R2 ;EXPECTED CONTENTS OF TSSR
3199            043730  032701  000100          BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
3200            043734  001402                  BEQ     25$             ;BRANCH IF NOT OFF-LINE
3201            043736  052702  000100          BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
3202            043742  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
3203            043744  001404                  BEQ     30$             ;OKAY IF MATCH
3207            043746                          ERRHRD  ERRNO,T112REJ,PKTSSR    ;COMMAND NOT REJECTED
                043746  104456                                                          TRAP    C$ERHRD
                043750  002127                                                          .WORD   1111
                043752  044632'                                                         .WORD   T112REJ
                043754  011656'                                                         .WORD   PKTSSR
3208            043756                  30$:
3209            043756  004737  010724'  35$:    JSR     PC,CKRAM        ;CHECK RAM TO MEMORY
3210            043762  103405                  BCS     59$             ;RAM OK GO ON
3214            043764                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
                043764  104456                                                          TRAP    C$ERHRD
                043766  002130                                                          .WORD   1112
                043770  004737'                                                         .WORD   PKTRAM
                043772  015320'                                                         .WORD   RAMERR
3215            043774                          ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                043774                                                          10000$:
                043774  104405                                                          TRAP    C$ESEG
3216
3217            043776                  59$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                043776                                                          L10104:
                043776  104403                                                          TRAP    C$ESUB
3218
3219                                    ;*
3220                                    ;
3221                                    ;TEST 11, SUBTEST 3
3222                                    ;
3223                                    ;CHECK THAT THE GET STATUS COMMAND IS ACCEPTED
3224                                    ;
3225                                    ;*
3226
3227            044000                          BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
                044000                                                          T11.3:
                044000  104402                                                          TRAP    C$BSUB
3228
3229            044002                          SETPRI  #PRI00          ;LOWER PRIORITY TO ALLOW INTERRUPTS
                044002  012700  000000                                                  MOV     #PRI00,R0
                044006  104441                                                          TRAP    C$SPRI
3230            044010                          BGNSEG                  ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
                044010  104404                                                          TRAP    C$BSEG
3231            044012  004737  015604'          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
3232            044016  103405                  BCS     5$              ;BR IF SOFT INIT = OK
3236            044020  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
```

```
3331 044330                            ERRDF   ERRNO,T11SR2,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     044330  104455                                                            TRAP    C$ERDF
     044332  002141                                                            .WORD   112'
     044334  045100'                                                           .WORD   T11SR2
     044336  011656'                                                           .WORD   PKTSSR
3332 044340                     15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044340  104406                                                            TRAP    C$CLP1
3333 044342                            ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     044342  104410                                                            TRAP    C$ESCAPE
     044344  000076                                                            .WORD   L10106-.
3334 044346  005737  002222'           TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
3335 044352  001004                    BNE     22$                     ;BRANCH IF YES
3339 044354                            ERRHRD  ERRNO,T11NINT,PKTSSR
     044354  104456                                                            TRAP    C$ERHRD
     044356  002142                                                            .WORD   1122
     044360  045154'                                                           .WORD   T11NINT
     044362  011656'                                                           .WORD   PKTSSR
3340 044364  016501  000002        22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
3341 044370  012702  100206           MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
3342 044374  032701  000100           BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
3343 044400  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
3344 044402  052702  000100           BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
3345 044406  020201        25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
3346 044410  001404                    BEQ     30$                     ;OKAY IF MATCH
3350 044412                            ERRHRD  ERRNO,T114REJ,PKTSSR    ;COMMAND NOT REJECTED
     044412  104456                                                            TRAP    C$ERHRD
     044414  002143                                                            .WORD   1123
     044416  044743'                                                           .WORD   T114REJ
     044420  011656'                                                           .WORD   PKTSSR
3351 044422                     30$:
3352
3353 044422  004737  010724'   35$:   JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
3354 044426  103404                    BCS     59$                     ;RAM OK GO ON
3358 044430                            ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     044430  104456                                                            TRAP    C$ERHRD
     044432  002144                                                            .WORD   1124
     044434  004737'                                                           .WORD   PKTRAM
     044436  015320'                                                           .WORD   RAMERR
3359 044440                     59$:
3360 044440                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<
     044440                                                                 10000$:
     044440  104405                                                            TRAP    C$ESEG
3361 044442                            ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\
     044442                                                                 L10106:
     044442  104403                                                            TRAP    C$ESUB
3362
3363 044444                            EXIT    TST                     ;ALL DONE THIS TEST
     044444  104432                                                            TRAP    C$EXIT
     044446  000772                                                            .WORD   L10102-.
3364
3365                            ;+
3366                            ;LOCAL STORAGE FOR THIS TEST
3367                            ;-
3368
3370 044450                            .BLKB   10-<.-TSV2E7>
3372 044460                     T11PACKET:                             ;COMMAND PACKET FOR TEST
3373 044460  100204                    .WORD   100204                  ;WRITE CHAR COMMAND, WITH IE, ACK
```

```
                                        .TITLE  TSV6 - PARAMETER CODING
                                        BGNMOD  TSV6
045442                          TSV6::
                                        .SBTTL  HARDWARE PARAMETER CODING SECTION
                                ;++
                                ; 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.
                                ;--
045442                                  BGN##RD
045442  000010                          .WORD L10107-L$HARD/2
045444                          L$HARD::
045444                                  GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
045444  000031                          .WORD   T$CODE
045446  045464'                         .WORD   HPM1
045450  160010                          .WORD   T$LOLIM
045452  177776                          .WORD   T$HILIM
045454                                  GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
045454  001031                          .WORD   T$CODE
045456  045520'                         .WORD   HPM2
045460  000000                          .WORD   T$LOLIM
045462  000776                          .WORD   T$HILIM
                                        ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
045464                                  ENDHRD
                                        .EVEN
045464                          L10107:
045464  104     105     126     HPM1:   .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
045520  111     116     124     HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
045544  111     116     124     HPM3:   .ASCIZ  'INTERRUPT PRIORITY      '
                                        .EVEN
                                        .SBTTL  SOFTWARE PARAMETER CODING SECTION
                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--
045574                                  BGNSFT
045574  000003                          .WORD L10110-L$SOFT/2
045576                          L$SOFT::
                                        GPRMA   SPM1,0,-1,YES                   ; GET TRANSPORT TEST FLAG.
045576                                  GPRMA   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
045576  001130                          .WORD   T$CODE
```

```
   045600  045634'                          .WORD   SPM4
   045602  177777                           .WORD   -1
56                              ;           GPRMD   SPM6,4,D,7777,0,7777,YES        ; GET LOCAL ERROR LIMIT
57                              ;           GPRMD   SPM7,6,D,7777,0,7777,YES        ; GET GLOBAL ERROR LIMIT
58 045604                                   ENDSET
                                            .EVEN
   045604                               L10110:
59
60
61 045604     105     116     101   SPM1:   .ASCIZ  'ENABLE TRANSPORT TESTS '
62 045634     111     116     110   SPM4:   .ASCIZ  'INHIBIT ITERATIONS     '
63                                  ;SPM6:  .ASCIZ  'PER TEST ERROR LIMIT   '
64                                  ;SPM7:  .ASCIZ  'PER UNIT ERROR LIMIT   '
65                                          .SBTTL  PATCH AREA
66
67                                  ;
68                                  ; FINALLY A GENEROUS PATCH AREA.
69                                  ;
70                                  ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
71                                  ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
72                                  ;
73
74 045664                           PATCH::
75
76                                  ;           .BLKW   32.
77 045664                                       .BLKW   1.
78
79                                  ;           .IF     NZ,.&377
80                                  ;           .=.!377+1
81                                  ;           .ENDC
82 045666                                       LASTAD          ;SET LAST USED ADDRESS.
                                                .EVEN
   045666  000000                               .WORD   0
   045670  000000                               .WORD   0
   045672                               L$LAST::
83 045672                                       ENDMOD
84                                              .SBTTL  HARD CODED P-TABLE
85                                  ;++
86                                  ;           DIAGNOSTIC IS PRE-PARAMETERIZED PER THIS TABLE
87                                  ;--
88 045672                                       BGNSETUP        1
89 045672                                       BGNPTAB
   045672  000000                               .WORD   0
   045674  000003                               .WORD   L10113-.,/2-1
   045676                               L10111:
90 045676  172522                               .WORD           172522
91 045700  000224                               .WORD           224
92 045702  000240                               .WORD           PRI05
93 045704                                       ENDPTAB
   045704                               L10113:
94 045704                                       ENDSETUP
95
96         000001                               .END
```

SYMBOL TABLE

```
T8NVCK  034637R     002 T92DAT  037052R     002 WF.IRE= 000040          XSOILC= 001000          X1.UNC= 000002
T8PACK  034540R     002 T92DON= 037066R     002 WF.IWF= 000020          XSOLET= 020000          X2.BUF= 000100
T8SSR   034730R     002 T92REJ  037141R     002 WF.IWR= 000100          XSOMOT= 000200          X2.EXT= 000200
T8VCK   034602R     002 T93REJ  037240R     002 WF.I3R= 000002          XSONEF= 002000          X2.OPM= 100000
T9      035036RG    002 T94REJ  037333R     002 WF.I4R= 000001          XSOONL= 000100          X2.RCE= 040000
T9BFR   037032R     002 T95REJ  037431R     002 WRTCHR  010472RG    002 XSOPED= 000010          X2.REV= 000077
T9DATA  037020R     002 UAM   = 000200 G         WRTERR  005105R     002 XSORLL= 010000          X2.SPA= 035400
T9INT   037707R     002 UNXIN   002200RG    002 WRTMSG  005050R     002 XSORLS= 040000          X2.UNI= 000007
T9LOOP  035060R     002 UNREC = 000006          WSMBK   021020RG    002 XSOTMK= 100000          X2.WCF= 002000
T9NBA   037066R     002 US    = 004115R     002 XFERAS  015550R     002 XSOVCK= 000020          X3.DCK= 000010
T9NINT  037616R     002 WAITF   016060RG    002 XNXM    016206R     002 XSOWLE= 004000          X3.MBZ= 000006
T9PACK  037010R     002 WC.IFA= 000200          XORBFO  007504R     002 XSOWLK= 000004          X3.MDE= 177400
T9REST  040106R     002 WC.IFE= 000002          XORFOR  007622R     002 XXCOMM  003120RG    002 X3.OPI= 000100
T9SSR   037527R     002 WC.IGO= 000001          XST0  = 000006 G          X$ALWA= 000000          X3.REV= 000040
T9TSBA  037776R     002 WC.IRE= 000010          XST1  = 000010 G          X$FALS= 000040          X3.RIB= 000001
T9.1    035060R     002 WC.IRW= 000004          XST2  = 000012 G          X$OFFS= 000400          X3.SPA= 000200
T9.2    035326R     002 WC.IOT= 000100          XST3  = 000014 G          X$TRUE= 000020          X3.TRF= 000020
T9.3    035540R     002 WC.I1T= 000040          XST4  = 000016 G          X1.COR= 020000          X4.HSP= 100000
T9.4    035744R     002 WC.I5R= 000020          XSOBOT= 000002            X1.DLT= 100000          X4.MBZ= 017400
T9.5    036132R     002 WF.IED= 000010          XSOEOT= 000001            X1.MBZ= 017375          X4.RCE= 040000
T9.6    036336R     002 WF.IER= 000004          XSOIE = 000040            X1.RBP= 000400          X4.TSM= 020000
T9.7    036570R     002 WF.IHI= 000200          XSOILA= 000400            X1.SPA= 040000          X4.WRC= 000377

. ABS.  000000      000
        000000      002
ABS     045704      002
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  28264 WORDS  ( 111 PAGES)
DYNAMIC MEMORY:  20614 WORDS  ( 79 PAGES)
ELAPSED TIME:  00:36:10
CZTSAA,CZTSAA.SEQ/-SP=SVC/ML,TSV1A,TSV22A,TSV3B,TSV4,TSV5A,TSV6
```