Micro Fiche Scan

Name of device(s) tested:
    TU80
Test description:
    TU80 FRONT END PRTB
MAINDEC Number or Package Identifier (after SEP 1977):
    CZTUXA0
Fiche Document Part Number:
    AH-T333A-MC
Fiche preparation date unknown, using copyright year:
    1983
Image resolution:
    8-bit gray levels, max. quality for archiving

.REM\

IDENTIFICATION

PRODUCT ID: AC-T332A-MC

PRODUCT TITLE: CZTUXA0 TU80 FRONT END PRT B

PRODUCT DATE: 23 - MARCH - 1983

MAINTAINER: TAPE DIAGNOSTIC ENGINEERING

AUTHOR: DICE SYSTEMS, INC.

TABLE OF CONTENTS

ABSTRACT

1.0 ABSTRACT

THIS IS A PDP-11 RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TU80 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A PDP-11 SYSTEM. THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. REFERENCE THE FOLLOWING DIGITAL EQUIPMENT DOCUMENTS:

1. ENGINEERING SPECIFICATION FOR TU80 MAGTAPE CONTROLLER; DOCUMENT NUMBER: YM-C194D-022; REVISION NUMBER 2; DATE: 28-JUL-81.

2. ENGINEERING SPECIFICATION FOR TU80 DIAGNOSTIC PACKAGE; DOCUMENT NUMBER: YM-C194F-00; REVISION NUMBER 0; DATE: 2-SEP-81.

3. ENGINEERING SPECIFICATION FOR TU80 MAGTAPE SUBSYSTEM; DOCUMENT NUMBER: YM-C194S-02; REVISION NUMBER 3; DATE: 10-JUN-81.

4. CIQPMA0 XXDP+ PROGRAMMER'S MANUAL; DOCUMENT NUMBER AC-S296A-AC DATE: 14 JULY 1980.

HARDWARE, SOFTWARE REQUIREMENTS AND PREREQUISITES

2.0 HARDWARE, SOFTWARE REQUIREMENTS AND PREREQUISITES

2.1 HARDWARE REQUIREMENTS

PDP-11 FAMILY PROCESSOR WITH 32K WORDS OF MEMORY
TU80 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE I.E. 4K FOR I/O PAGE)

2.2 OPTIONAL HARDWARE:

UP TO 4 TU80 CONTROLLERS PER PDP-11 UP TO 1 DRIVES PER CONTROLLER

2.3 SOFTWARE REQUIREMENTS

PDP-11 DIAGNOSTIC SUPERVISOR (HSAAD0.SYS)
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

2.4 PREREQUISITES

FUNCTIONAL PDP-11 FAMILY CENTRAL PROCESSOR AND MEMORY
FUNCTIONAL CONSOLE TERMINAL
FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR

OPERATING INSTRUCTIONS - OPERATOR COMMANDS

3.0 OPERATING INSTRUCTIONS

3.1 OPERATOR COMMANDS

THE TU80 DIAGNOSTIC IS A PDP-11 DIAGNOSTIC SUPERVISOR COMPATIBLE PROGRAM. ALL LOADING AND RUN TIME INSTRUCTIONS CAN BE REFERENCED IN THE PDP-11 PROGRAMMER'S MANUAL "CIQPMA0 XXDP+ PROGRAMMER'S MANUAL, NUMBER AC-S296A-AC.

BOOT THE DIAGNOSTIC XXDP MEDIA

```
CHMDLB0 XXDP+ DL MONITOR 28K
BOOTED VIA UNIT 0

ENTER DATE (DD-MMM-YY): 29-JAN-82
RESTART ADDRESS:  153726
50 HZ ? N <CR>
LSI ? N Y
THIS IS XXDP+ TYPE "H" OR "H/L" FOR DETAILS

R CZTUXA0

CZTUXABINDRS LOADED
DIAG. RUN-TIME SERVICES REV. D APR 79
CZTUX-A-0
****TU80 LOGIC DIAGNOSTIC****
UNIT IS TU80
DR>
DRS>START/FLAG:PNT:HOE
```

THE ABOVE COMMANDS WILL START THE DIAGNOSTIC. THE COMMAND HAS TWO SWITCHES ON WHICH ARE "PRINT EACH TEST NBR AS EXECUTED" AND "HALT ON ERROR".

3.2 HARDWARE PARAMETERS

AFTER INITIAL STARTING OF THE PROGRAM (START COMMAND TO THE DIAGNOSTIC SUPERVISOR), THE PROGRAM WILL ISSUE THE "CHANGE HW?" QUESTION TO ASK IF THE HARDWARE PARAMETERS ARE TO BE CHANGED (BY THE OPERATOR).

ON A "N" (NO) RESPONSE TO THE "CHANGE HW?" QUESTION, THE DIAGNOSTIC WILL NOT RUN. IT WILL GIVE THE MESSAGE "NO UNIT". A "Y" IS REQUIRED AND AT LEAST A "1" IS REQUIRED AT THE "# UNITS (D)?" QUESTION.

TSBA/TSDB = 172522, VECTOR = 224

ON A "Y" (YES) RESPONSE TO THE QUESTION, THE FOLLOWING QUESTIONS WILL THEN BE ASKED TO ALLOW THE OPERATOR TO SELECT THE UNITS TO BE TESTED. A VALUE, IF PRESENT, LOCATED TO THE LEFT OF THE QUESTION MARK IS THE

OPERATING INSTRUCTIONS

DEFAULT VALUE THAT WILL BE TAKEN IF ONLY A CARRIAGE RETURN IS TYPED AS A RESPONSE. A "(D)" IN A QUESTION INDICATES THAT A DECIMAL NUMBER IS REQUIRED AS A RESPONSE. AN "(O)" INDICATES AN OCTAL NUMBER IS BEING SOLICITED. AN "(L)" INDICATES THAT A LOGICAL RESPONSE IS TO BE MADE "Y" FOR YES, "N" FOR NO.

```
# UNITS (D) ?  <ENTER THE NUMBER OF M7454 CONTROLLERS
                PRESENT TO BE TESTED>

UNIT 0

DEVICE ADDRESS  (O) 172522 ? <ENTER THE ADDRESS OF THE
                              TSSR REGISTER>

VECTOR (O) 224 ?         <ENTER ADDRESS OF INTERRUPT
                          VECTOR>
```

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE "# UNITS?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING.

3.3 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED ON A START, RESTART, OR CONTINUE; THEY ALLOW FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES.

```
CHANGE SW (L) ? <TYPE Y TO CAUSE THE FOLLOWING
                 QUESTIONS TO BE ASKED>

INHIBIT ITERATIONS (L) N ?  <TYPE "Y" TO PREVENT MULTIPLE
                             ITERATIONS OF CERTAIN TESTS.
                             THIS CAUSES EACH TEST PASS TO
                             RUN AS QUICKLY AS POSSIBLE.
                             ONLY QUICK-RUNNING LOGIC
                             TESTS USE MULTIPLE
                             ITERATIONS.>
```

OPERATING INSTRUCTIONS - SAMPLE PRINTOUTS

4.0 OPERATING INSTRUCTIONS - SAMPLE PRINTOUTS

4.1 SUCCESSFUL RUN EXAMPLE (PDP-11)

```
TST: 001 FIFO EXERCISER TEST
TST: 002 INITIALIZE #4 TEST
TST: 003 OFF-LINE REJECT AND REWIND TEST
TST: 004 BASIC WRITE DATA TEST
TST: 005 BASIC READ DATA (FORWARD AND REVERSE) TEST
TST: 006 MANUAL INTERVENTION TEST
TST: 007 CONFIGURATION TYPEOUT TEST
TST: 008 SCOPE LOOPS TEST


0 ERRORS

NOTE: PROGRAM NOW STARTS OVER AGAIN AT TEST 1
```

OPERATING INSTRUCTIONS - SAMPLE ERROR MESSAGES

5.0 OPERATING INSTRUCTIONS - SAMPLE ERROR MESSAGES

ERROR MESSAGE EXAMPLE 1

```
TST: 001 FIFO EXERCISER TEST
CZTUX HRD ERR 01610 ON UNIT 00 TST 016 SUB 002 PC: 040624
FIFO STATUS (IN WORD  9) INCORRECT AFTER WRITE FIFO

 TAPE BUS SIGNALS IN WORD #8: - DESIGNATOR <BIT #>
PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>
IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>
IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>
 TAPE BUS SIGINALS IN WORD #9:
DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>
 MESSAGE BUFFER ADDRESS = 047352
 MESSAGE BUFFER CONTENTS:
WORD #0 EXPD: 100020  RECV: 100020  XOR: 000000
WORD #1 EXPD: 000012  RECV: 000012  XOR: 000000
WORD #2 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #3 EXPD: 000010  RECV: 000010  XOR: 000000
WORD #4 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #5 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #6 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #7 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #8 EXPD: 070217  RECV: 070217  XOR: 000000
WORD #9 EXPD: 000074  RECV: 000034  XOR: 000040
```

ERROR MESSAGE EXAMPLE 2

```
CZTUX HRD ERR 00159 ON UNIT 00 TST 001 SUB 005 PC: 026202
TSSR NOT CORRECT AFTER SPACE RECORDS COMMAND
 TSSR = 100214
 TSSR BITS SET: SC,SSR
 TERMINATION CLASS CODE = UNRECOVERABLE ERROR
 PACKET ADDRESS = 026420
 PACKET WORD #  = 140010
 PACKET WORD #  = 000010
 PACKET WORD #  = 000000
 PACKET WORD #  = 000024
```

ERROR MESSAGE EXAMPLE 3

```
CZTX HRD ERR 00121 ON UNIT 00 TST 001 SUB 002 PC: 023306
MOT BIT (XSTO) NOT SET DURING REWIND (EXTENDED FEATURES MODE)
 EXPD: 000312 RECV: 000112 XOR: 000200
```

PROGRAM RUN TIMES

6.0 PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAM ARE LISTED BELOW.THESE FIGURES ARE TO BE USED AS A GUIDE. THE TIMING WAS DONE ON A PDP-11/34 PROCESSOR WITH A LA-34 CONSOLE.

THE PROGRAM RUNS IN TWO MODES: NO ITERATIONS AND DEFAULT MODE. IN THE NO ITERATIONS MODE, EACH TEST IS RUN ONCE, WITH NO ITERATIONS. IN THE DEFAULT MODE EACH TEST IS REPEATED BY THE NUMBER OF TIMES INDICATED BY THE ITERATION COUNT. NO ITERATIONS MODE IS SELECTED BY ANSWERING THE INHIBIT ITERATIONS QUESTION WITH A "Y" (YES).

TEST NUMBER	N/I SECS.	ITER SECS	DEF SECS.
1	1	1	0
2	1	1	0
3	1	1	0
4	1	1	0
5	1	1	0
6	N/A		
7	N/A		
8	N/A		

THE TIMES REQUIRED TO RUN TESTS 1 THROUGH 5 IN ONE COMMAND:

Q.V.	2 MINS 19 SECONDS
DEFAULT	11 MINS 35 SECONDS

7.0 TEST SUMMARIES

7.1 TEST 1 - FIFO EXERCISER

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S *
* CONTROLLER (M7454)                                                 *
**********************************************************************
```

THIS TEST USES THE WRITE SUBSYSTEM MEMORY COMMAND TO VERIFY THE CONTROLLER'S FIFO AND ASSOCIATED STATUS AND CONTROL LOGIC.

7.1.1 TEST 1, SUBTEST 1:-

THIS TEST VERIFIES, BY USING THE READ STATUS SELECT CODE, THAT THE FIFO STATUS IS IN THE CORRECT INITIAL STATE AFTER THE CONTROLLER IS INITIALIZED (INPUT READY TRUE, OUTPUT READY AND DATA IN MISS FALSE). THESE STATUS SIGNALS ARE CHECKED BY THE CONTROLLER'S SELF-TEST SEQUENCE, SO THIS SUBTEST IS ACTUALLY MORE OF A PARTIAL CHECK OF THE READ STATUS FUNCTION THAN THE FIFO STATUS.

7.1.2 TEST 1, SUBTEST 2:-

THIS SUBTEST VERIFIES THE ABILITY OF THE FIFO TO CORRECTLY PASS A DATA BYTE FROM INPUT TO OUTPUT. FOR EACH OF THE 256 DATA VALUES (0-377 OCTAL) THE FOLLOWING IS DONE:

1. INITIAL FIFO STATUS IS CHECKED
2. THE WRITE FIFO FUNCTION, SPECIFYING A COUNT OF ONE BYTE TO BE WRITTEN, IS EXECUTED.
3. READ STATUS IS EXECUTED AND FIFO STAUS IS CHECKED.
4. READ FIFO IS EXECUTED AND THE DATA AND FINAL STATUS IS CHECKED.

7.1.3 TEST 1, SUBTEST 3:-

THIS SUBTEST VERIFIES THE ABILITY OF THE FIFO TO CORRECTLY PASS MULTIPLE DATA BYTES FROM INPUT TO OUTPUT. THE FOLLOWING SEQUENCE IS DONE WITH VARIOUS DATA PATTERNS AND BYTE COUNTS FROM 2 TO 64.

1. INITIAL FIFO STATUS IS CHECKED.
2. THE WRITE FIFO FUNCTION.
3. RESD STATUS IS EXECUTED AND FIFO STATUS IS CHECKED.
4. READ FIFO IS EXECUTED AND THE DATA AND FINAL STAUS IS CHECKED.

7.1.4 TEST 1, SUBTEST 4;-

THIS SUBTEST VERIFIES THAT READING THE FIFO WHEN IT IS EMPTY CAUSES THE LAST WORD (ILW) STATUS TO ASSERT.

7.1.5 TEST 1, SUBTEST 5:-

THIS SUBTEST VERIFIES THAT WRITING 64. BYTES INTO FIFO, WITHOUT READING ANY OUT, CAUSES THE INPUT READY STATUS TO NEGATE. THE SUBTEST THEN VERIFIES THAT WTITING A 65TH BYTE INTO FIFO CAUSES THE DATA IN MISS STATUS TO ASSERT. NEXT IT IS VERIFIED THAT THE ORIGINAL 64 BYTES CAN BE READ OUT CORRECTLY AND THAT THE DATA HAS NOT BEEN CORRUPTED.

7.1.6 TEST 1 SUBTEST 6:-

THIS SUBTEST VERIFIES THAT THE RESET FIFO FUNCTION WITHIN THE WRITE MISCELLANIOUS CONTROL 1 FUNCTION INITIALIZES THE FIFO TO THE CORRECT INITIAL STATUS. THE FOLLOWING STEPS ARE PERFORMED:

1. RESET AN ALREADY INITIALIZED FIFO AND CHECK FOR PROPER STATUS.
2. WRITE A VARYING NUMBER OF BYTES (1-65.) INTO THE FIFO AND VERIFY THAT AFTER EACH BLOCK OF BYTES IS WRITTEN THE FIFO CAN BE RESET TO ITS INITIAL STATE.

7.2 TEST 2 - INITIALIZE #4

```
**************************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TU80'S                 *
* CONTROLLER (M7454)                                                     *
**************************************************************************
```

THIS TEST VERIFIES THAT WRITING INTO THE TSSR RETURNS THE CONTROLLER TO ITS INITIALIZED STATE FROM VARIOUS CONDITONS.

7.3 TEST 3 - OFF LINE REJECT AND REWIND TEST

THIS TEST VERIFIES BASIC TAPE MOTION COMMAND DECODING AND BASIC OPERATION OF THE REWIND POSITIONING COMMAND. IT DOES NOT NECESSARILY DEMONSTRATE THAT THE TRANSPORT AN BE REWOUND FROM AN ARBITRARY POSITION ON THE TAPE. SUBSEQUENT TESTS IMPLICITLY CHECK THE OPERATION OF THE REWIND COMMAND SINCE THEY MUST TYPICALLY REWIND THE TAPE IN IN THE NORMAL COURSE OF THEIR TEST SEQUENCES. THE TEST CONSISTS OF THE FOLLOWING THREE SUBTESTS:

7.3.1 TEST 3, SUBTEST 1:-

THIS SUBTEST VERIFIES THAT ALL TAPE MOTION COMMANDS (WITH

VALID MODE CODES) TERMINATE WITH FUNCTION REJECT AND OFF LINE STATUS WHEN THE TAPE TRANSPORT IS OFF-LINE.

7.3.2 TEST 3, SUBTEST 2:-

THIS SUBTEST VERIFIES THAT A REWIND COMMAND, WITH THE CLEAR VOLUME CHECK (CVC) BIT CLEAR IS REJECTED IF THE VOLUME CHECK (VCK) FLAG IS SET.

7.3.3 TEST 3, SUBTEST 3:-

THIS SUBTEST VERIFIES THAT A REWIND COMMAND WITH A CVC=1 CLEARS VCK AND RETURNS PROPER STATUS IN THE MESSAGE BUFFER.

7.4 TEST 4 - BASIC WRITE DATA TEST

```
**************************************************************************
* NOTE: THIS TEST MUST HAVE A GOOD MAGTAPE IN THE DRIVE ANY              *
* TAPE ERRORS WILL BE DISPLAYED AS TAPE STATUS ALERT                     *
**************************************************************************
```

THIS TEST VERIFIES THAT THE WRITE DATA (NEXT) COMMAND OPERATES CORRECTLY, UP TO THE POINT OF CHECKING THAT THE DATA WAS ACTUALLY WRITTEN ONTO THE TAPE CORRECTLY. THE TESTING IN THIS TEST IS LIMITED TO VERIFYING THAT THE COMMAND WAS TERMINATED CORRECTLY WITH THE CORRECT REGISTER, BUFFER, AND RAM CONTENTS.

7.4.1 TEST 4, SUBTEST 1:-

THIS SUBTEST VERIFIES THAT A WRITE COMMAND (ANY VALID MODE CODE) WITH THE CLEAR VOLUME CHECK (CVC) BIT CLEAR IS REJECTED IF THE VOLUME CHECK (VCK) FLAG IS SET. ALL VALID MODE CODES ARE CHECKED (WRITE DATA, WRITE RETRY).

7.4.2 TEST 4, SUBTEST 2:-

THIS SUBTEST VERIFIES THAT WRITE DATA COMMANDS WITH CVC=1 AND THE SWAP BYTES (SWB) BIT CLEAR OPERATES PROPERLY. THE BYTE COUNT (RECORD SIZE) VARIES FROM 20 THROUGH 64K IN VARYING INCREMENTS (DEPENDING ON WHETHER OR NOT THE DIAGNOSTIC IS RUNNING ON THE LONG VERIFICATION MODE). THE TAPE IS NOT REWOUND BETWEEN SUCCESSIVE RECORDS BUT IS REWOUND AFTER THE FINAL RECORD IS WRITTEN. AN INCREMENTING COUNT PATTERN IS SUPPLIED IN THE DATA BUFFER. AFTER EACH BLOCK IS WRITTEN, THE TTSR AND THE TSBA REGISTERS AND THE MESSAGE BUFERS ARE CHECKED.

7.4.3 TEST 4, SUBTEST 3:-

THIS SUBTEST VERIFIES THAT WRITE DATA COMMANDS WITH CVC=1 AND

THE SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST SEQUENCE
IS IDENTICAL TO THAT USED IN SUBTEST 2. THE RESULTS SHOULD BE
THE SAME.

7.4.4 TEST 4, SUBTEST 4;-

THIS SUBTEST VERIFIES THAT A WRITE COMMAND WITH AN ILLEGAL
BUFFER ADDRESS IS REJECTED WITH THE PROPER ERROR STATUS
AND THAT TAPE DOES NOT MOVE.

7.4.5 TEST 4, SUBTEST 5:-

THIS SUBTEST VERIFIES THAT A WRITE DATA COMMAND SPECIFYING
A DATA BUFFER STARTING IN NONEXISTANT MEMORY TERMINATES WITH
THE PROPER ERROR STATUS WITHOUT MOVING TAPE. THIS TEST IS
SKIPPED IF NONEXISTANT MEMORY CAN NOT BE ADDRESSED.

7.4.6 TEST 4, SUBTEST 6:-

THIS SUBTEST VERIFIES THAT A WRITE DATA COMMAND SPECIFYING A
DATA BUFFER IN STARTING IN EXISTANT MEMORY BUT RUNNING INTO NONEXISTANT
MEMORY TERMINATES WITH THE PROPER ERROR STATUS. A LARGE
ENOUGH RECORD SIZE IS SPECIFIED SUCH THAT TAPE IS ACTUALLY
MOVED AND WRITTEN.

7.5 TEST 5 - BASIC READ DATA TEST (FORWARD AND REVERSE)

```
**************************************************************************
* NOTE: THIS TAPE MUST HAVE A GOOD MAGTAPE IN THE DRIVE                  *
* ANY TAPE ERRORS WILL BE DISPLAYED AS TAPE STATUS ALERT                 *
**************************************************************************
```

THIS TEST VERIFIES THAT THE READ FORWARD AND READ REVERSE COMMANDS
OPERATE PROPERLY. VARIOUS COMBINATIONS OF ODD AND EVEN DATA
BUFFER BOUNDARIES, RECORD SIZES AND BYTE SWAP CONTROL VARIABLES
ARE USED. THE TEST FURTHER VERIFIES THE WRITE DATA COMMAND
BY ACTUALLY READING AND VERIFYING WRITTEN DATA. ALSO TESTED
ARE PROPER TERMINATIONS ON EXCEPTIONAL OR ERROR CONDITIONS:
RECORD LENGTH LONG, RECORD LENGTH SHORT, READ REVERSE AT BOT,
ILLEGAL DATA BUFFER ADDRESSES, AND NONEXISTANT DATA BUFFER
ADDRESSES.

7.5.1 TEST 5, SUBTEST 1:-

THIS SUBTEST VERIFIES THAT THE READ FORWARD COMMAND WITH SWB=0
OPERATES PROPERLY. THE TAPE IS FIRST REWOUND AND THEN WRITTEN
WITH A SERIES OF TEST RECORDS VARYING IN LENGTH AND DATA

CONTENT. THE TAPE IS THEN REWOUND AGAIN AND THE RECORD READ SEQUENTIALLY AND RESULTS (STATUS, DATA, ETC.) VERIFIED. THE BYTE COUNT ON EACH READ FORWARD COMMAND IS SET TO THE LENGTH OF THE EXPECTED RECORD, SO NO EXCEPTIONAL CONDITIONS SHOULD OCCUR.

7.5.2 TEST 5, SUBTEST 2:-

THIS SUBTEST VERIFIES THAT THE RAD DATA COMMANDS WITH CVC=1 AND THE SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2. THE RESULTS, EXCEPT FOR RAM CONTENTS SHOULD BE THE SAME.

7.5.3 TEST 5, SUBTEST 3:-

THIS SUBTEST VERIFIES THAT A READ FORWARD COMMAND READING A RECORD LONGER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE STATUS ALERT TERMINATION WITH THE RECORD LENGTH (RLL) BIT SET.

7.5.4 TEST 5, SUBTEST 4;-

THIS SUBTEST VERIFIES THAT A READ FORWARD COMMAND READING A RECORD SHORTER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE STATUS ALERT TERMINATION WITH THE RECORD LENGTH SHORT (RLS) BIT SET. IT IS VERIFIED THAT THE RESIDUAL BYTE COUNT (RBPCR) IN THE MESSAGE BUFFER CONTAINS THE PROPER NONZERO VALUE (E.G. THE DIFFERENCE BETWEEN THE ACTUAL BYTE COUNT AND THE ACTUAL RECORD LENGTH).

7.5.5 TEST 5, SUBTEST 5:-

OPERATES PROPERLY. THE TAPE IS FIRST REWOUND AND THEN WRITTEN WITH A SERIES OF TEST RECORDS VARYING IN LENGTH AN DATA CONTENT. THE TAPE IS THEN READ IN REVERSE SEQUENTIALLY AND AND THE RESULTS (STATUS, DATA, ETC.) VERIFIED. THE BYTE COUNT ON EACH READ REVERSE COMMAND IS SET TO THE LENGTH OF THE EXPECTED RECORD, SO NO EXCEPTIONAL CONDITIONS SHOULD OCCUR.

7.5.6 TEST 5, SUBTEST 6:-

THIS SUBTEST VERIFIES THAT THE READ DATA COMMANDS WITH CVC=1 AND THE SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2. THE RESULTS EXCEPT FOR RAM CONTENTS SHOULD BE THE SAME.

7.5.7 TEST 5, SUBTEST 7:-

THIS SUBTEST VERIFIES THAT A READ REVERSE COMMAND, READING A RECORD LONGER THAN THE SPECIFIED BYTE COUNT, CAUSES A TAPE STATUS ALERT TERMINATION WITH THE RECORD LENGTH LONG (RLL) BIT SET.

7.5.8 TEST 5, SUBTEST 8:-

THIS SUBTEST VERIFIES THAT A READ REVERS COMMAND SPECIFYING A DATA BUFFER STARTING IN NONEXISTANT MEMORY TERMINATES WITH THE PROPER ERROR STATUS WITHOUT MOVING THE TAPE.

7.5.9 TEST 5, SUBTEST 9:-

THIS SUBTEST VERIFIES THAT ILLEGAL BUFFER ADDRESSES CAUSE A FUNCTION REJECT TERMINATION WITH ILLEGAL ADDRESS (ILA) ERROR BIT SET.

7.5.10 TEST 5, SUBTEST 10:-

THIS SUBTEST VERIFIES THAT A DATA BUFFER ADDRESS, REFERENCING NONEXISTANT MEMORY, CAUSES RECOVERABLE ERROR TERMINATION (TC=4), WITH THE NXM BIT SET IN THE TSSR, AND THAT THE TAPE IS ULTIMATELY POSITIONED PROPERLY.

7.5.11 TEST 5, SUBTEST 11:-

THIS SUBTEST VERIFIES THAT A READ REVERSE COMMAND ISSUED WHILE THE TAPE IS AT BOT RESULTS IN A FUNCTION REJECT TERMINATION WITH THE NONEXECUTABLE FUNCTION (NEF) ERROR BIT SET.

7.5.12 TEST 5, SUBTEST 12:-

THIS SUBTEST VERIFIES THAT A READ REVERSE COMMAND ISSUED WHILE THE TAPE IS POSITIONED BEFORE THE FIRST RECORD ON TAPE (BUT NOT AT BOT) RESULTS IN TAPE STATUS ALERT.

7.6 TEST 6 - MANUAL INTERVENTION

THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST") THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF THE SUBSYSTEM THAT CAN NOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE. THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR ARE BY TYPING <CTRL-C> OR BY SELECTING CODE 3. SELECTION CODES AND SUBROUTINES ARE:

CODE	ROUTINE
0	HELP. PRINTS THIS MENU
1	OFFLINE/ONLINE ATTENTION TEST
2	WRITE-PROTECT TEST

3 EXIT (RETURN TO SUPERVISOR)

EACH MENU ITEM CORRESPONDS TO A SUBTEST AS FOLLOWS:

PRINTS OUT THE MENU ON THE CONSOLE TERMINAL.

THIS ROUTINE INITIALIZES THE CONTROLLER, ISSUES A WRITE CHARACTER-
ISTICS COMMAND TO ENABLE ATTENTION INTERRUPTS, ISSUES A MESSAGE
BUFFER RELEASE COMMAND, PRINTS A MESSAGE ON THE CONSOLE TERMINAL
INSTRUCTING THE THE OPERATOR TO TOGGLE THE ON-LINE SWITCH ON THE
TRANSPORT, THEN WAITS FOR AN ATTENTION INTERRUPT. EACH TIME THE TRANSPORT
TRANSITIONS FROM ON-LINE TO OFF-LINE OR VICE VERSA, AN ATTENTION INTERRUPT
SHOULD BE GENERATED. THE PROGRAM WILL REPORT THE INTERRUPT AND THE
THE CURRENT STATE OF THE TRANSPORT. THE OPERATOR SHOULD VERIFY THAT
THE REPORTED STATE MATCHES THE STATE INDICATED BY THE LED ON THE
FRONT PANEL OF THE TRANSPORT. IN ADDITION, WHEN THE TRANSPORT IS PLACED
OFF-LINE, THE PROGRAM ISSUES A SEQUENCE OF TAPE MOTION COMMANDS
(READ, WRITE, POSITION, ETC.) AND VERIFIES THAT, FOR EACH
COMMAND, FUNCTION REJECT TERMINATION RESULTS, ALONG WITH THE
NON-EXECUTABLE FUNCTION (NEF) ERROR BIT BEING SET.

THIS ROUTINE INSTRUCTS THE OPERATOR TO MOUNT A SCRATCH TAPE REEL
THAT DOES NOT HAVE A WRITE ENABLE RING INSTALLED, THEN WAITS
FOR THE OPERATOR TO RESPOND THAT THIS HAS BEEN ACCOMPLISHED.
UPON THE RESPONSE, THE PROGRAM VERIFIES THAT THE TRANSPORT
SHOWS A WRITE-PROTECTED STATUS, THEN ATTEMPTS TO WRITE DATA
ON THE TAPE AND EXPECTS THE APPROPRIATE ERROR TERMINATION
INDICATING THAT THE WRITE FUNCTION COULD NOT BE PERFORMED
BECAUSE THE REEL IS WRITE PROTECTED. IF THE APPROPRIATE
TERMINATION IS NOT RECEIVED, AN ERROR IS REPORTED.

7.8 TEST 8 - CONFIGURATION TYPEOUT

THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE
TERMINAL THE CONFIGURATION OF THE M7454 MODULE AND THE TU80
SUBSYSTEM. SPECIFICALLY, THE FOLLOWING INFORMATION IS PRINTED:

1. MICROCODE REVISION LEVEL OF THE M7454.
2. NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.
3. UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE
   ENABLED/PROTECTED)OF EACH CONNECTED TRANSPORT.

THE OPERATOR IS EXPECTED TO READ THE PRINTOUT AND VERIFY THAT IT
MATCHES THE ACTUAL CONFIGURATION AT HAND. IF, FOR EXAMPLE, THE
PROGRAM INDICATES THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN
FACT ONLY ONE IS PRESENT, THE OPERATOR MUST INTERPRET THIS AS AN
ERROR AND ATTEMPT TO FIND THE CAUSE (BAD CABLE, FAULTY UNIT SELECT
DECODING IN THE TRANSPORT, ETC.).[SINCE THE CONTROLLER CAN ONLY
ACCESS UNIT 0 IF IT IS IN "STANDARD" MODE, THE PROGRAM WILL FORCE THE
MODULE INTO EXTENDED MODE VIA THE WRITE SUBSYSTEM MEMORY COMMAND
IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]

THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO

THE DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE SELECTED (E.G. FROM THE INITIAL STARTUP DIALOGUE), THE ROUTINE WILL BE REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.

7.8 TEST 8 - SCOPE LOOPS

THIS IS A STANDALONE ROUTINE PROVIDING A NUMBER OF TIGHT "SCOPE LOOPS" USEFUL FOR DEBUGGING BASIC REGISTER ACCESS PROBLEMS WITH THE M7454 MODULE. THESE SCOPE LOOPS CAN BE USED WHEN THE NORMAL "LOOP ON ERROR" OR "LOOP ON TEST (SUBTEST)" FACILITIES DON'T SEEM TO ALLOW THE OPERATOR TO ZERO IN A PROBLEM IN THE EARLY TESTS (I.E. THE HARDWARE MAY NOT BE RESPONDING TO A REGISTER ACCESS, CAUSING A BUS ERROR TRAP, EVEN THOUGH THE DEVICE ADDRESS SELECTED BY THE PROGRAM MATCHES THE THE CONFIGURATION SET UP IN THE HARDWARE DIP SWITCHES). THE FOLLOWING MENU OF SCOPE LOOPS IS AVAILABLE:

CODE	SCOPE LOOP
0	HELP. PRINT THIS MENU
1	TSBA READ ACCESS
2	TSSR READ ACCESS
3	INITIALIZE (TSSR WRITE ACCESS)
4	TSDB HIGH BYTE WRITE ACCESS
5	TSDB LOW BYTE WRITE ACCESS
6	TSDB MAINTENANCE-MODE WORD WRITE ACCESS
7	EXIT (RETURN TO SUPERVISOR)

FOR SCOPE LOOPS THAT WRITE INTO REGISTERS, THE PROGRAM PROMPTS THE OPERATOR FOR THE DATA TO BE WRITTEN. TYPING <RETURN> CAUSES AN EXIT FROM THE SCOPE LOOP BACK TO THE MENU.

```
                                        .SBTTL  PROGRAM HEADER
                                        .MCALL  SVC
000000                                  SVC                     ; INITIALIZE SUPERVISOR MACROS
                                        .ENABLE LC
                                        .NLIST  BEX,CND
000000                                  .ENABL  AMA,ABS
        002000                          . = 2000
002000                                  BGNMOD  TUV2A
002000                          TUV2A::

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


002000                                  POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT,BGNSETUP
002000                                  HEADER  CZTUX,A,0,655.,0
002000                          L$NAME::                ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  132                             .ASCII /Z/
002002  124                             .ASCII /T/
002003  125                             .ASCII /U/
002004  130                             .ASCII /X/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                          L$REV::                 ;REVISION LEVEL
002010  101                             .ASCII  /A/
002011                          L$DEPO::                ;0
002011  060                             .ASCII  /0/
002012                          L$UNIT::                ;NUMBER OF UNITS
002012  000001                          .WORD   T$PTHV
002014                          L$TIML::                ;LONGEST TEST TIME
002014  001217                          .WORD   655.
002016                          L$HPCP::                ;POINTER TO H.W. QUES.
002016  066646                          .WORD   L$HARD
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  067006                          .WORD   L$SOFT
002022                          L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002124                          .WORD   L$HW
002024                          L$SPTP::                ;PTR. TO S.W. PTABLE
002024  002134                          .WORD   L$SW
002026                          L$LADP::                ;DIAG. END ADDRESS
002026  067220                          .WORD   L$LAST
002030                          L$STA::                 ;RESERVED FOR APT STATS
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                          .WORD   0
002036                          L$APT::                 ;APT EXPANSION
002036  000000                          .WORD   0
002040                          L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  067174                          .WORD   L$DISPATCH
002042                          L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
```

PROGRAM HEADER

```
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  003334                  .WORD   L$DVTYP
002062                  L$REPP::                ;PTR. TO REPORT CODE
002062  023046                  .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  022544                  .WORD   L$AU
002072                  L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022642                  .WORD   L$DU
002074                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                  .WORD   0
002076                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  003342                  .WORD   L$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  021746                  .WORD   L$INIT
002106                  L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  023024                  .WORD   L$CLEAN
002110                  L$ACP::                 ;PTR. TO AUTO CODE
002110  022750                  .WORD   L$AUTO
002112                  L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021736                  .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
```

```
                                        .SBTTL  DEFAULT HARDWARE P-TABLE

                                ;++
                                ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                                ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
                                ;--
002122                                  BGNHW   DFPTBL          ;DEFAULT HARD-P-TABLE
002122  000003                          .WORD   L10000-L$HW/2
002124                          L$HW::
002124                          DFPTBL::

002124  172522                          .WORD   172522          ; 2ND (OF 2) REGISTERS.
002126  000224                          .WORD   224             ; INTERRUPT VECTOR
002130  000240                          .WORD   PRI05           ; INTERRUPT PRIORITY.
002132                                  ENDHW
002132                          L10000:
```

```
                                .SBTTL  SOFTWARE P-TABLE

                               ;++
                               ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
                               ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                               ;--
002132                                 BGNSW   SFPTBL
002132  000004                        .WORD   L10001-L$SW/2
002134                         L$SW::
002134                         SFPTBL::

002134  000000                 TRANSTST::     .WORD   0       ;ENABLE RAM DUMP
002136  000000                 NOITS::        .WORD   0       ; INHIBIT ITERATION OPTION.
                                                              ; ... 0 = ITERATE.
                                                              ; ...NZ = INHIBIT ITERATE.
002140  000031                 LERRMAX::      .WORD   25.     ; LOCAL (PER TEST) ERROR LIMIT
002142  000310                 GERRMAX::      .WORD   200.    ; GLOBAL (PER UNIT) ERROR LIMIT
002144                                 ENDSW
002144                         L10001:

```

```
                                .SBTTL  GLOBAL EQUATES SECTION


                                .SBTTL  GLOBAL EQUATES SECTION

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

                                       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
                               ;
```

```
                                PRI07== 340
                                PRI06== 300
                                PRI05== 240
                                PRI04== 200
                                PRI03== 140
                                PRI02== 100
                                PRI01== 40
                                PRI00== 0
                                ;
                                ;OPERATOR FLAG BITS
                                ;
                                EVL==       4
                                LOT==      10
                                ADR==      20
                                IDU==      40
                                ISR==     100
                                UAM==     200
                                BOE==     400
                                PNT==    1000
                                PRI==    2000
                                IXE==    4000
                                IBE==   10000
                                IER==   20000
                                LOE==   40000
                                HOE==  100000

929
930 002144                      KT11    ,,      ;DEFINE MEMORY MANAGEMENT REGISTERS
                                .SBTTL  MEMORY MANAGEMENT DEFINITIONS
                                ;*KT11 VECTOR ADDRESS
                                MMVEC=  250
                                ;*KT11 STATUS REGISTER ADDRESSES
                                SR0=    177572
                                SR1=    177574
                                SR2=    177576
                                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 REGISTORS
                                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
```

```
                                .SBTTL  TU80 REGISTER AND PACKET DEFINITIONS

                        ;
                        ; SOME GENERAL EQUATES.
                        ;

000004                  ERRVEC==        4               ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
000060                  TTIVEC==        60              ; INTERRUPT VECTOR FOR CONSOLE INPUT
177560                  TTICSR==        177560          ; BUS ADDRESS OF CONSOLE INPUT
177562                  TTIBFR==        177562          ; CONSOLE INPUT DATA BUFFER

                        ;+
                        ;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                  XS0TMK= BIT15                   ;TAPE MARK DETECTED
040000                  XS0RLS= BIT14                   ;RECORD LENGTH SHORT
020000                  XS0LET= BIT13                   ;LOGICAL END OF TAPE
010000                  XS0RLL= BIT12                   ;RECORD LENGTH LONG
004000                  XS0WLE= BIT11                   ;WRITE LOCK ERROR
002000                  XS0NEF= BIT10                   ;NON EXECUTABLE FUNCTION
001000                  XS0ILC= BIT9                    ;ILLEGAL COMMAND
000400                  XS0ILA= BIT8                    ;ILLEGAL ADDRESS
000200                  XS0MOT= BIT7                    ;TAPE IN MOTION
000100                  XS0ONL= BIT6                    ;TRANSPORT ON LINE
000040                  XS0IE=  BIT5                    ;INTERRUPT ENABLE
000020                  XS0VCK= BIT4                    ;VOLUME CHECK BIT
000010                  XS0PED= BIT3                    ;PHASE ENCODED DRIVE
000004                  XS0WLK= BIT2                    ;WRITE LOCKED
000002                  XS0BOT= BIT1                    ;BEGINNING OF TAPE
000001                  XS0EOT= BIT0                    ;END OF TAPE


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

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

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
                        ;(XST2)
                        ;-
                        X2.OPM  = BIT15                 ;OPERATION IN PROGRESS (TAPE MOVING)
                        X2.RCE  = BIT14                 ;RAM CHECKSUM ERROR
                        X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8   ;NOT USED BY TU80 (ALWAYS=0)
                        X2.WCF  = BIT10                 ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
                        X2.EXTF = BIT7                  ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
                        X2.BUFE = BIT6                  ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
                        X2.REV  = 000077                ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
                        X2.UNIT = BIT2+BIT1+BIT0        ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
                        ;(XST3)
                        ;-
                        X3.MDE  = 177400                ;MICRO-DIAGNOSTIC ERROR CODE
                        X3.SPARE= BIT7                  ;NOT USED BY TU80
                        X3.OPI  = BIT6                  ;OPERATION INCOMPLETE
                        X3.REV  = BIT5                  ;REVERSE
                        X3.TRF  = BIT4                  ;TRANSPORT RESPONSE FAILURE
                        X3.DCK  = BIT3                  ;DENSITY CHECK
                        X3.MBZ  =BIT2+BIT1              ;NOT USED ALWAYS 0
                        X3.RIB  = BIT0                  ;REVERSE INTO BOT

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


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

                                TSREJ=  3*2                     ;COMMAND REJECTED
                                UNREC=  6                       ;UNRECOVERABLE ERROR

                        ;+
                        ;
                        ;DEVICE REGISTER OFFSETS
                        ;
```

```
                                ;-
                                
177776                          TSBA==  -2
177776                          TSBAL== -2
177776                          TSDB==  -2                      ;TSDB/TSBA REGISTER
177776                          TSDBL== -2                      ;TSDB/TSBA REGISTER
177777                          TSBAH== -1
177777                          TSDBH== -1                      ;TSDB/TSBA REGISTER HIGH BYTE
000000                          TSSR==  0                       ;TSSR REGISTER
000001                          TSSRH== 1                       ;TSSR 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
000020                          RMPKTBEG= 20                    ;COMMAND PACKET BEGIN RAM ADDRESS
000027                          RMPKTEND= 27                    ;COMMAND PACKET END RAM ADDRESS
000104                          RMMSGBEG= 104                   ;MESSAGE BUFFER BEGIN RAM ADDRESS
```

```
1106    000117                  RMMSGEND= 117                           ;MESSAGE BUFFER END RAM ADDRESS
1107                            ;+
1108                            ;
1109                            ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
1110                            ;
1111                            ;-
1112
1113    000006                  XST0==  6                               ;EXTENDED STATUS REGISTER 0 (WORD 4)
1114    000010                  XST1==  8.                              ;EXTENDED STATUS REGISTER 1 (WORD 5)
1115    000012                  XST2==  10.                             ;EXTENDED STATUS REGISTER 2 (WORD 6)
1116    000014                  XST3==  12.                             ;EXTENDED STATUS REGISTER 3 (WORD 7)
1117    000016                  XST4==  14.                             ;EXTENDED STATUS REGISTER 4 (WORD 8)
1118
1119
1120                            ;+
1121                            ;
1122                            ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
1123                            ;
1124                            ;-
1125
1126    000002                          PKLOW   = 2                     ;LOW ORDER CHARACTERISTIC DATA POINTER
1127    000004                          PKHI    = 4                     ;HIGH ORDER CHARACTERISTIC DATA POINTER
1128    000006                          PKBCNT  = 6                     ;NUMBER OF BYTES IN DATA PACKET
1129
1130    000010                          EXBCNT=10                       ;NUMBER OF BYTES IN EXTENDED DATA PACKET
1131
1132                            ;+
1133                            ;DATA PACKET OFFSETS FOR WRITE SUBSYTEM COMMAND
1134                            ;-
1135    000000                          BSEL0   = 0                     ;BYTE 0
1136    000001                          BSEL1   = 1                     ;BYTE 1
1137    000002                          SEL2    = 2                     ;WORD 2
1138    000004                          SELDATA = 4                     ;WORD 3
1139
1140                            ;+
1141                            ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
1142                            ;-
1143    000000                  PW.NOP          = 0                     ;NO-OP
1144    000001                  PW.RDRAM        = 1                     ;READ RAM
1145    000002                  PW.WTRAM        = 2                     ;WRITE RAM
1146    000003                  PW.RFIFO        = 3                     ;READ FIFO
1147    000004                  PW.WFIFO        = 4                     ;WRITE FIFO
1148    000005                  PW.RDSTAT       = 5                     ;READ STATUS
1149    000006                  PW.WCTL         = 6                     ;WRITE TAPE CONTROL
1150    000007                  PW.WFMT         = 7                     ;WRITE TAPE FORMAT
1151    000010                  PW.WMISC        = 10                    ;WRITE MISCELLANEOUS
1152    000011                  PW.WNPR         = 11                    ;WRITE NPR CONTROL
1153    000020                  PW.D22          = 20                    ;DO MICROTEST 22
1154    000021                  PW.D11          = 21                    ;DO MICROTEST 11
1155    000022                  PW.D13          = 22                    ;DO MICROTEST 13
1156    000023                  PW.NO1311       = 23                    ;DISABLE MICROTEST 11 AND 13
1157    000024                  PW.RDEXT        = 24                    ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPO
1158
1159                            ;+
1160                            ;BSEL1 CODES FOR WRITE TAPE CONTROL
1161                            ;-
1162    000200                  WC.IFAD         = BIT7                  ;IFAD - FORMATTER ADDRESS
```

```
1163        000100              WC.IOTAD        = BIT6              ;ITAD0  - TRANSPORT ADDRESS BIT 0
1164        000040              WC.I1TAD        = BIT5              ;ITAD1  - TRANSPORT ADDRESS BIT 1
1165        000020              WC.I5RESV       = BIT4              ;IRESV5 - RESERVED #5
1166        000010              WC.IREW         = BIT3              ;IREW   - REWIND
1167        000004              WC.IRWU         = BIT2              ;IRWU   - REWIND AND UNLOAD
1168        000002              WC.IFEN         = BIT1              ;IFEN   - FORMATTER ENABLE
1169        000001              WC.IGO          = BIT0              ;GO
1170
1171                            ;+
1172                            ;BSEL1 CODES FOR WRITE FORMAT
1173                            ;-
1174        000200              WF.IHISP        = BIT7              ;IHISP  - HIGH SPEED
1175        000100              WF.IWRT         = BIT6              ;IWRT   - WRITE
1176        000040              WF.IREV         = BIT5              ;IREV   - REVERSE
1177        000020              WF.IWFM         = BIT4              ;IWFM   - WRITE FILE MARK
1178        000010              WF.IEDIT        = BIT3              ;IEDIT  - EDIT
1179        000004              WF.IERASE       = BIT2              ;IERASE - ERASE
1180        000002              WF.I3RESV       = BIT1              ;IRESV3 - RESERVED #3
1181        000001              WF.I4RESV       = BIT0              ;IRESV4 - RESERVED #4
1182
1183
1184                            ;+
1185                            ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
1186                            ;-
1187        000200              MS.EXT          = BIT7              ;INVERT SENSE OF EXTENDED FEATURES SWITCH
1188        000020              MS.RSFIFO       = BIT4              ;RESET FIFO AND INPUT PARITY ERRORR
1189        000010              MS.RSTAPE       = BIT3              ;RESET TAPE STATUS IN 2 FLIP-FLOPS
1190        000006              MS.ATTN         = BIT2!BIT1         ;ATTENTION TRIGGER FIELD
1191        000001              MS.RSD          = BIT0              ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
1192                            ;+
1193                            ; MS.ATTN SUBCODES
1194                            ;-
1195        000000                    MSA.NOP = 0*2                 ;NO-OP (NOTHING TRIGGERED)
1196        000002                    MSA.VOL = 1*2                 ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
1197        000004                    MSA.NRAM= 2*2                 ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
1198        000006                    MSA.FRAM= 3*2                 ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
1199                            ;+
1200                            ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
1201                            ;-
1202        000200              NP.IR           = BIT7              ;INTERRUPT REQUEST (0-1 TRANSITION)
1203        000100              NP.OUT          = BIT6              ;TAPE DATA DIRECTION OUT (0= IN)
1204        000040              NP.LOOP         = BIT5              ;ENABLE TRANSPORT LOOPBACK
1205        000020              NP.WRP          = BIT4              ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
1206                            ;+
1207                            ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
1208                            ;-
1209
1210        000200              S2.DIM          = BIT7              ;WORD #9 BYTE 2 DATA IN MISS
1211        000100              S2.ILW          = BIT6              ;               ILW H
1212        000040              S2.OUTRDY       = BIT5              ;               OUT RDY H
1213        000020              S2.INRDY        = BIT4              ;               IN RDY H
1214        000010              S2.ATIMR        = BIT3              ;               TIMER A FLAG H
1215        000004              S2.BTIMR        = BIT2              ;               TIMER B FLAG H
1216        000003              S2.UNDEF        = BIT1+BIT0         ;(UNDEFINED)
1217        100000              S1.PARIN        = BIT15             ;WORD #8 BYTE 1 PARIN H
1218        040000              S1.I2RESV       = BIT14             ;               IRESV2
1219        020000              S1.I1RESV       = BIT13             ;               IRESV1
```

```
                                S1.IEOT         = BIT12                 ;                       IEOT L
                                S1.IIDENT       = BIT11                 ;                       IIDENT H
                                S1.ICER         = BIT10                 ;                       ICER H
                                S1.IFMK         = BIT9                  ;                       IFMK H
                                S1.IHER         = BIT8                  ;                       IHER H
                                S0.ISPEED       = BIT7                  ;WORD #8 BYTE 0         ISPEED H
                                S0.IRDY         = BIT6                  ;                       IRDY L
                                S0.IONL         = BIT5                  ;                       IONL L
                                S0.ILDP         = BIT4                  ;                       ILDP L
                                S0.IDBY         = BIT3                  ;                       IDBY L
                                S0.IRWD         = BIT2                  ;                       IRWD L
                                S0.IFBY         = BIT1                  ;                       IFBY L
                                S0.IFPT         = BIT0                  ;                       IFPT L
                        ;       SPECIAL KEYBORD STUFF FOR MOVER PROGRAM

                                TKS     =177560                         ;KEYBOARD STATUS REGISTER
                                TKB     =177562                         ;KEYBOARD DATA REGISTER
                                TPS     =177564                         ;CONSOLE PRINTER STATUS REGISTER
                                TPB     =177566                         ;CONSOLE PRINTER DATA REGISTER
                                HIMEM   =007776                         ;HIGH MEMORY MASK VALUE
                        ;       CONTROLLER DEFINITIONS
                                CSR     =174400                         ;STATUS AND CONTROL REGISTER
                                BAR     =174402                         ;DL ADDRESS REGISTER
                                DAR     =174404                         ;PLATTER ADDRESS
                                MPR     =174406                         ;MULTIPURPOSE REGISTER
                        ;
                        ;       CONTROLLER COMMANDS
                        ;
                                DLGETS  =4                              ;GET STATUS COMMAND
                                SEEK    =6                              ;SEEK TRACK AND HEAD SELECT
                                DLRDHD  =10                             ;READ SECTOR HEADER
                                READ    =14                             ;READ COMMAND
                                DLRDNH  =16                             ;READ SECTOR NO HEADER CHECK
                        ;
                                READY   =1                              ;DRIVE READY BIT IN STATUS REG.
                                DLSR    =13                             ;STATUS AND RESET
                                DLERR   =177730                         ;MASK FOR COVER OPEN
                                DLUN    =6                              ;HEADS UNLOADED
                                DLCYL   =000177                         ;MASK FOR CYLINDER ADDRESS
                                DLDNER  =100200                         ;DONE SET OR ERROR SET BITS
                        ;
                                TTICSR  =       177560                  ;KEYBOARD INPUT STATUS
                                TTIBFR  =       177562                  ;KEYBOARD DATA REGISTER
                                TTOCSR  =       177564                  ;CONSOLE PRINTER STATUS REGISTER
                                TTOBFR  =       177566                  ;CONSOLE PRINTER DATA REGISTER
```

```
                                .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,B
                                        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.
                                ;

002144  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.
                                ;
002146  000000                  EPRTSW::        .WORD   0               ;PRINT SWITCH
002150  000000                  UNITN::         .WORD   0               ;UNIT # UNDER TEST.
002152  000000                  QVP::           .WORD   0               ;QUICK VERIFY FLAG.
002154  000000                  CSRADDR::       .WORD   0               ;ADDRESS OF CSR FOR CURRENT DEVICE
002156  000224                  IVEC::          .WORD   224             ;INTERRUPT VECTOR
002160  000200                  IPRI::          .WORD   PR104           ;INTERRUPT PRIORITY.
002162  000000                  TSTCNT::        .WORD   0               ;NUMBER OF TESTS RUN IN THIS PASS
002164  000000                  LOOPCNT::       .WORD   0               ;REMAINING ITERATION COUNT FOR TEST
002166  000000                  DEVCNT::        .WORD   0               ;NUMBER OF DEVICE UNDER TEST
002170  000000                  FATFLG::        .WORD   0               ;SET IF FATAL ERROR IS DETECTED IN TEST
002172  000000                  INTRECV::       .WORD   0               ;SET IF TAPE INTERRUPT WAS RECEIVED
002174  000000                  BENBSW::        .WORD   0               ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
002176  000000                  EXPD::          .WORD   0               ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
002200  000000                  RECV::          .WORD   0               ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
002202  000000                  ERRHI::         .WORD   0               ;HIGH ADDRESS MEMORY ERROR
002204  000000                  ERRLO::         .WORD   0               ;LOW ADDRESS MEMORY ERROR
002206                          RAMDATA::       .BLKW   16.             ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
002246  000000                  RAMSIZ::        .WORD   0               ;RAM DATA SIZE FOR PRAMPKT ROUTINE
002250  000000                  RCVHIADD::      .WORD   0               ;RECEIVED BUFFER HIGH ADDRESS
002252  000000                  RCVLOADD::      .WORD   0               ;RECEIVED BUFFER LOW ADDRESS
002254  000000                  COUNT::         .WORD   0               ;TEST COUNT PATTERN
002256  000000                  DATA::          .WORD   0               ;TEST DATA
002260  000000                  TSTFLAG::       .WORD   0               ;TEST FLAG WORD
002262  000000                  TSTPTR::        .WORD   0               ;TSTBLK POINTER
002264  000000                  PRMNO::         .WORD   0               ;PRINT ROUTINE TEMP
002266                          EXPMSG::        .BLKB   100.            ;EXPECTED MESSAGE BUFFER DATA
002432                          RECMSG::        .BLKB   100.            ;RECEIVED MESSAGE BUFFER DATA
002576                          TMPBFR::        .BLKB   80.             ;TEMPORARY STORAGE FOR PRINT
002716  000000                  MESBFA::        .WORD   0               ;STORES ADDRESS OF MESSAGE BUFFER FOR ERR PRT
```

```
                                        .SBTTL  TSTBLK  - TEST DATA TABLE

                                ;+
                                ;
                                ;THIS TABLE CONTAINS TEST DATA USED IN SEVERAL TESTS
                                ;
                                ;IN SEQUENCE THE DATA IS:
                                ;
                                ;       ALL ZEROS
                                ;       ALL ONES
                                ;       WALKING ONES
                                ;       WALKING ZEROS
                                ;       ALTERNATING ONES AND ZEROS
                                ;
                                ;-

002720                          TSTBLK::
002720  000000                          .WORD   0                       ;ALL ZEROS
002722  177777                          .WORD   177777                  ;ALL ONES
002724  000001                          .WORD   BIT0                    ;DATA FOR WALKING ONES
002726  000002                          .WORD   BIT1
002730  000004                          .WORD   BIT2
002732  000010                          .WORD   BIT3
002734  000020                          .WORD   BIT4
002736  000040                          .WORD   BIT5
002740  000100                          .WORD   BIT6
002742  000200                          .WORD   BIT7
002744  000400                          .WORD   BIT8
002746  001000                          .WORD   BIT9
002750  002000                          .WORD   BIT10
002752  004000                          .WORD   BIT11
002754  010000                          .WORD   BIT12
002756  020000                          .WORD   BIT13
002760  040000                          .WORD   BIT14
002762  100000                          .WORD   BIT15
002764  177776                          .WORD   ^CBIT0                  ;DATA FOR WALKING ZEROS
002766  177775                          .WORD   ^CBIT1
002770  177773                          .WORD   ^CBIT2
002772  177767                          .WORD   ^CBIT3
002774  177757                          .WORD   ^CBIT4
002776  177737                          .WORD   ^CBIT5
003000  177677                          .WORD   ^CBIT6
003002  177577                          .WORD   ^CBIT7
003004  177377                          .WORD   ^CBIT8
003006  176777                          .WORD   ^CBIT9
003010  175777                          .WORD   ^CBIT10
003012  173777                          .WORD   ^CBIT11
003014  167777                          .WORD   ^CBIT12
003016  157777                          .WORD   ^CBIT13
003020  137777                          .WORD   ^CBIT14
003022  077777                          .WORD   ^CBIT15
003024  125252                          .WORD   125252                  ;ALTERNATING ONES, ZEROS
003026  052525                          .WORD   052525                  ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
003030                          TBLEND==.
```

```
                                        .SBTTL  GLOBAL ENVIRONMENT STORAGE
                                ;
                                ;STORAGE FOR DEVICE REGISTERS
                                ;
003030  000000  100000  000000 DUMMY:  0,100000,0,0                  ;DUMMY DEVICE REGISTERS...
003040  000000  000000  000000        0,0,0,0,0,0,0,0
                                                                    ;...FOR MULTI-UNIT CHECKOUT.


003060  000000                 DUFLG::         .WORD   0           ;"DROPPED UNIT" FLAG.
                                                                    ;INHIBITS CODE IN "CLEAN-UP".
003062  000000                 NODEV::         .WORD   0           ;FLAG TO SAY NO DEVICE.

003064  000000                 TEMP1::         .WORD   0           ;SOME TEMP LOCATIONS.
003066  000000                 TEMP2::         .WORD   0
003070  000000                 XXCOMM::        .WORD   0           ;XXDP+ COMM BLOCK POINTER.
003072  000000                 FREE::          .WORD   0           ;1ST FREE MEMORY ADDRESS...
003074  000000                 FRESIZ::        .WORD   0           ;...AND SIZE (IN WORDS).
003076  000000                 FREEHI:         .WORD   0           ;LAST WORD IN FREE SPACE
003100  000000                 KTFLG::         .WORD   0           ;KT11, MEM AVAIL FLAG -
                                                                    ;- .WORD        0 = <24K OR NO KT -
                                                                    ;- NZ = >24K AND KT.

003102  000000                 KTENABLE::      .WORD   0           ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
003104  002000                 PST32W::        .WORD   2000        ;32W BLOCK ADDRESS FOR 32K START
003106  000000                 SIFLAG::        .WORD   0           ;
003110  000000                 BADDAT::        .WORD   0           ;ACTUAL DATA
003112  000000                 GDDAT::         .WORD   0           ;EXPECTED DATA
003114  000000                 LOOPFL::        .WORD   0
003116                         CTAB::                              ;CONFIGURATION TABLES.
003116  000000                 CTABM::         .WORD   0           ;CONFIG WORK.
003120  000000                                 .WORD   0
003122  000000                                 .WORD   0
003124  000000                                 .WORD   0
003126  177777                                 .WORD   -1          ;END OF MEM TABLE.
003130                         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
                                ;
003130                         ERTABL:         .BLKW   64.
003330  000000                 ERTABE:         .WORD   0

003332  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
                                ;-

003334                                  DEVTYP  <TU80>
003334                          L$DVTYP::
003334  124  125  070                    .ASCIZ  /TU80/
                                        .EVEN

                                ;+
                                ;TEST DESCRIPTION
                                ;-
003342                                  DESCRIPT <CZTUXA0 TU80 FRONT END PRT B>
003342                          L$DESC::
003342  103  132  124                    .ASCIZ  /CZTUXA0 TU80 FRONT END PRT B/
                                        .EVEN
                                ;+
                                ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
                                ;-

003400  003440  003443  003447  TSSRBIT::          .WORD   1$,2$,3$,4$,5$,6$,7$,8$
003420  003501  003505  003511                    .WORD   9$,10$,11$,12$,13$,14$,15$,16$
003440    123    103    000    1$:      .ASCIZ  'SC'
003443    102    111    105    2$:      .ASCIZ  'BIE'
003447    123    103    105    3$:      .ASCIZ  'SCE'
003453    122    115    122    4$:      .ASCIZ  'RMR'
003457    116    130    115    5$:      .ASCIZ  'NXM'
003463    116    102    101    6$:      .ASCIZ  'NBA'
003467    102    111    124    7$:      .ASCIZ  'BIT9'
003474    102    111    124    8$:      .ASCIZ  'BIT8'
003501    123    123    122    9$:      .ASCIZ  'SSR'
003505    117    106    114    10$:     .ASCIZ  'OFL'
003511    102    111    124    11$:     .ASCIZ  'BIT5'
003516    102    111    124    12$:     .ASCIZ  'BIT4'
003523    102    111    124    13$:     .ASCIZ  'BIT3'
003530    102    111    124    14$:     .ASCIZ  'BIT2'
003535    102    111    124    15$:     .ASCIZ  'BIT1'
003542    102    111    124    16$:     .ASCIZ  'BIT0'
                                        .EVEN
003550    124    123    123    SFIERR:  .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
003603    124    123    123    SFHERR:  .ASCIZ  'TSSR ERROR AFTER BUS RESET'
003636    040    040    116    NXR:     .ASCIZ  /  NON-EXISTANT DEVICE REGISTER/
003675    045    101    040    NXRX:    .ASCIZ  /%A  ADDRESS: %O6/
003716    045    101    040    TSSX:    .ASCII  /%A  TSBA,TSSR EXP'D: %O6%A,%O6%N/
003756    045    101    040             .ASCIZ  /%A  TSBA,TSSR REC'D: %O6%A,%O6/
004015    045    116    045    FUSI:    .ASCII  /%N%A/
004021    040    040    125    USI:     .ASCIZ  /  UNEXPECTED INTERRUPT/
004050    040    040    111    NSI:     .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
```

```
1545 004113  045  116  045  FNOINTR:          .ASCII  /%N%A/
1546 004117  040  040  116  NOINTR: .ASCIZ   /  NO INTERRUPT WAS GENERATED/
1547 004154  040  040  111  IFAULT: .ASCIZ   /  INTERRUPT FAULT/
1548 004176  045  101  040  INTX:   .ASCIZ   /%A  CPU PC: %O6%A  TSBA: %O6/
1549 004233  040  040  042  NOINIT: .ASCIZ   /  "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
1550 004305  040  040  042  NSINIT: .ASCIZ   /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
1551 004355  040  040  042  BRINIT: .ASCIZ   /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/
1552
1553 004425  000            NUL:    .ASCIZ   //
1554 004426  045  116  000  NULCR:  .ASCIZ   /%N/
1555 004431  045  101  040  EXPGOT: .ASCIZ   /%A EXP'D: %O6%A, REC'D: %O6/
1556 004465  045  116  045  EXPGT2: .ASCIZ   /%N%A EXP'D: %O6%A, %O6%N%A REC'D: %O%A, %O6/
1557 004541  045  101  040  DUAD12: .ASCIZ   /%A REG(W) WRITTEN TO: %O6%A REG(R) READ; EXP'D: %O6%A, REC'D: %O6/
1558 004643  122  101  115  PKTRAM::         .ASCIZ  'RAM Contents Do Not Match Packet Sent'
1559 004711  040  040  103  SCME:   .ASCIZ   /  CONFIG DOESN'T MATCH MFG. MASTER/
1560 004754  127  122  111  WRTMSG: .ASCIZ   'WRITE CHARACTERISTICS Failed'
1561 005011  124  123  123  WRTERR: .ASCIZ   'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
1562 005104  124  123  123  RDERR:  .ASCIZ   'TSSR Incorrect After READ Command, More Bits Set Than SSR'
1563                         .EVEN
1564
1565
1566
```

```
                                        .SBTTL  GLOBAL ERROR REPORT SECTION

                                ;++
                                ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
                                ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
                                ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
                                ;--

005176                                  BGNMSG  NXRERR                  ;NON-EXISTANT DEVICE REGISTER.
005176                          NXRERR::
005176                                  PRINTX  #NXRX,NODEV             ;NODEV = NEXM ADDRESS.
005176  013746  003062                    MOV     NODEV,-(SP)
005202  012746  003675                    MOV     #NXRX,-(SP)
005206  012746  000002                    MOV     #2,-(SP)
005212  010600                            MOV     SP,R0
005214  104415                            TRAP    C$PNTX
005216  062706  000006                    ADD     #6,SP
005222  004737  005230                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQUIRED.
005226                                  ENDMSG
005226                          L10002:
005226  104423                          TRAP    C$MSG


                                ;
                                ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
                                ; TO ANY OF THE ABOVE ERROR SIGNATURES.
                                ;
005230  005727                  EXTEND: TST     (PC)+
005232  000000                  EXTA:   0                               ; 0 = NO EXTENSION.
005234  001402                          BEQ     1$
005236  004777  177770                  JSR     PC,@EXTA                ; APPEND EXTENSION TEXT.
005242                          1$:     PRINTX  #NULCR                  ; PRINT A BLANK LINE
005242  012746  004426                    MOV     #NULCR,-(SP)
005246  012746  000001                    MOV     #1,-(SP)
005252  010600                            MOV     SP,R0
005254  104415                            TRAP    C$PNTX
005256  062706  000004                    ADD     #4,SP
005262  000207                          RTS     PC
```

```
                                        .SBTTL  PRITSSR - PRINT TSSR CONTENTS

                                ;+
                                ;
                                ;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
                                ;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
                                ;BY A MESSAGE PRINTING ROUTINE
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      CONTENTS OF TSSR
                                ;
                                ;SUBORDINATE ROUTINES:
                                ;
                                ;       CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
                                ;
                                ;-

005264                          PRITSSR:
005264                                  SAVREG                          ;SAVE GENERAL REGISTERS
005270  010104                          MOV     R1,R4                   ;SAVE THE TSSR CONTENTS
005272                          PRINTB  #TSSRFOR,R4                     ;PRINT THE CONTENTS OF TSSR
005272  010446                          MOV     R4,-(SP)
005274  012746  006105                  MOV     #TSSRFOR,-(SP)
005300  012746  000002                  MOV     #2,-(SP)
005304  010600                          MOV     SP,R0
005306  104414                          TRAP    C$PNTB
005310  062706  000006                  ADD     #6,SP
005314  010400                          MOV     R4,R0                   ;GET TSSR BACK FOR CHKAMB
005316  004737  016700                  JSR     PC,CHKAMB               ;ARE CONTENTS AMBIGUOUS ?
005322  103410                          BCS     5$                      ;BRANCH IF NOT
005324                          PRINTX  #AMBTSSR                        ;SHOW CONTENTS ARE AMBIGUOUS
005324  012746  006325                  MOV     #AMBTSSR,-(SP)
005330  012746  000001                  MOV     #1,-(SP)
005334  010600                          MOV     SP,R0
005336  104415                          TRAP    C$PNTX
005340  062706  000004                  ADD     #4,SP
005344  010403                  5$:     MOV     R4,R3                   ;CONTENTS OF TSSR
005346  042703  001476                  BIC     #HIADDR!FATERR!TERCLS,R3        ;CLEAR ALL MULTIPLE BIT FIELDS
005352  001434                          BEQ     20$                     ;NO BITS ARE SET
005354  012702  002576                  MOV     #TMPBFR,R2              ;TEMPORARY ASCII BUFFER
005360  012701  003400                  MOV     #TSSRBIT,R1             ;ASCII EQUIVALENT OF BITS
005364  005703                  10$:    TST     R3                      ;REMAINING BITS TO CONVERT
005366  001413                          BEQ     15$                     ;BRANCH WHEN ALL ARE DONE
005370  000241                          CLC                             ;CLEAR CARRY FOR SHIFT
005372  006103                          ROL     R3                      ;SHIFT NEXT BIT TO CARRY
005374  103006                          BCC     13$                     ;BRANCH IF BIT NOT SET
005376  011100                          MOV     (R1),R0                 ;POINTER TO BIT DEFINITION
005400  112022                  11$:    MOVB    (R0)+,(R2)+             ;MOVE ASCIZ TO BUFFER
005402  001376                          BNE     11$                     ;MOVE ALL BITS
005404  112762  000054  177777          MOVB    #',,-1(R2)              ;INSERT A COMMA TO TERMINATE
005412  005721                  13$:    TST     (R1)+                   ;POINT TO NEXT DESCRIPTION
005414  000763                          BR      10$                     ;GET THE REMAINING BITS
005416  105042                  15$:    CLRB    -(R2)                   ;TERMINATE THE LINE
005420                          PRINTX  #TSSDEF,#TMPBFR                 ;PRINT THE BIT DEFINITIONS
005420  012746  002576                  MOV     #TMPBFR,-(SP)
005424  012746  006276                  MOV     #TSSDEF,-(SP)
```

```
                005430  012746  000002                  MOV     #2,-(SP)
                005434  010600                          MOV     SP,R0
                005436  104415                          TRAP    C$PNTX
                005440  062706  000006                  ADD     #6,SP
     1638
     1639       005444  010403                  20$:    MOV     R4,R3                   ;GET THE TSSR CONTENTS
     1640       005446  042703  177761                  BIC     #^CTERCLS,R3            ;CLEAR ALL BUT TERMINATION
     1641       005452  016303  006366                  MOV     TCOCOD(R3),R3           ;GET THE TERMINATION CODE MEANING
     1642       005456                                  PRINTX  #TCOASC,R3              ;PRINT THE TERMINATION CODE
                005456  010346                          MOV     R3,-(SP)
                005460  012746  006166                  MOV     #TCOASC,-(SP)
                005464  012746  000002                  MOV     #2,-(SP)
                005470  010600                          MOV     SP,R0
                005472  104415                          TRAP    C$PNTX
                005474  062706  000006                  ADD     #6,SP
     1643       005500  010403                          MOV     R4,R3                   ;TSSR CONTENTS AGAIN
     1644       005502  042703  177717                  BIC     #^CFATERR,R3            ;CLEAR ALL BUT FATAL TERMINATION
     1645       005506  001421                          BEQ     25$                     ;DON'T PRINT IF ZERO
     1646       005510  006203                          ASR     R3
     1647       005512  006203                          ASR     R3
     1648       005514  006203                          ASR     R3                      ;ALINE TERMINATION CODE FOR INDEX
     1649       005516  016303  006726                  MOV     TSFCOD(R3),R3           ;GET THE FATAL TERMINATION CODE
     1650       005522                                  PRINTX  #TFCASC,R3              ;PRINT THE FATAL TERMINATION CODE
                005522  010346                          MOV     R3,-(SP)
                005524  012746  006227                  MOV     #TFCASC,-(SP)
                005530  012746  000002                  MOV     #2,-(SP)
                005534  010600                          MOV     SP,R0
                005536  104415                          TRAP    C$PNTX
                005540  062706  000006                  ADD     #6,SP
     1651       005544  012737  000031  002170          MOV     #25.,FATFLG             ;DROP THIS UNIT AFTER ERROR
     1652       005552  010403                  25$:    MOV     R4,R3                   ;GET TSSR CONTENTS
     1653       005554  042703  176377                  BIC     #^CHIADDR,R3            ;CLEAR ALL BUT EXTENDED ADDRESS
     1654       005560  001411                          BEQ     30$                     ;DON'T PRINT IF ZERO
     1655       005562                                  PRINTX  #TEXASC,R3              ;PRINT THE EXTENDED ADDRESS BITS
                005562  010346                          MOV     R3,-(SP)
                005564  012746  006125                  MOV     #TEXASC,-(SP)
                005570  012746  000002                  MOV     #2,-(SP)
                005574  010600                          MOV     SP,R0
                005576  104415                          TRAP    C$PNTX
                005600  062706  000006                  ADD     #6,SP
     1656       005604  022704  100210          30$:    CMP     #100210,R4              ;CHECK FOR MEDIA ERROR
     1657       005610  001003                          BNE     31$                     ;BR, IF PROBABLY NOT TAPE ERROR
     1658       005612  012737  006014  002146          MOV     #EPRT3,EPRTSW           ;"PROBABLY MEDIA RELETED ERROR - BAD TAPE"
     1659       005620  005737  002146          31$:    TST     EPRTSW                  ;CHECK FOR THE SWITCH EMPTY
     1660       005624  001003                          BNE     310$                    ;BR, IF SWITCH IS NOT EMPTY
     1661       005626  012737  005755  002146          MOV     #EPRT1,EPRTSW           ;SET SWITCH TO DEFAULT
     1662       005634  013737  002146  005644  310$:   MOV     EPRTSW,32$+2            ;PUT REAL SWITCHABLE MESSAGE IN PLACE
     1663       005642                          32$:    PRINTB  #EPRT1                  ;PRINT THE ERROR MESSAGE
                005642  012746  005755                  MOV     #EPRT1,-(SP)
                005646  012746  000001                  MOV     #1,-(SP)
                005652  010600                          MOV     SP,R0
                005654  104414                          TRAP    C$PNTB
                005656  062706  000004                  ADD     #4,SP
     1664       005662  012737  005755  002146          MOV     #EPRT1,EPRTSW           ;RESET TO NORMAL ERROR POINTER
     1665       005670  000207                          RTS     PC                      ;RETURN TO CALLER
     1666       005672      045     116     045  EPRT2:  .ASCIZ  '%N%A *****CHECK M7454, CABLES AND TRANSPORT*****%S'
     1667       005755      045     116     045  EPRT1:  .ASCIZ  '%N%A *****REPLACE M7454*****%S'
```

```
1668 006014      045      116      045  EPRT3:  .ASCIZ  '%N%A *****POSSIBLE MEDIA RELATED ERROR - BAD TAPE*****%S'
1669
1670 006105      045      116      045  TSSRFOR:        .ASCIZ  '%N%A TSSR = %O6'
1671 006125      045      116      045  TEXASC:         .ASCIZ  '%N%A Extended Address Bits = %O6'
1672 006166      045      116      045  TCOASC:         .ASCIZ  '%N%A Termination Class Code = %T'
1673 006227      045      116      045  TFCASC:         .ASCIZ  '%N%A Fatal Termination Class Code = %T'
1674 006276      045      116      045  TSSDEF:         .ASCIZ  '%N%A TSSR Bits Set: %T'
1675 006325      045      116      045  AMBTSSR:        .ASCIZ  '%N%A TSSR Contents Are Ambiguous'
1676                                    .EVEN
1677 006366  006406  006431  006457  TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
1678 006406      116      157      162  1$:     .ASCIZ  'Normal Termination'
1679 006431      124      145      162  2$:     .ASCIZ  'Termination Condition'
1680 006457      124      141      160  3$:     .ASCIZ  'Tape Status Alert'
1681 006501      106      165      156  4$:     .ASCIZ  'Function Reject'
1682 006521      122      145      143  5$:     .ASCIZ  'Recoverable Error - Tape Position One Record Down'
1683 006603      122      145      143  6$:     .ASCIZ  'Recoverable Error - Tape Was Not Moved'
1684 006652      125      156      162  7$:     .ASCIZ  'Unrecoverable Error'
1685 006676      106      141      164  8$:     .ASCIZ  'Fatal Controller Error'
1686                                    .EVEN
1687
1688 006726  006736  006772  007003  TSFCOD: .WORD   1$,2$,3$,4$
1689 006736      111      156      164  1$:     .ASCIZ  'Internal Diagnostic Failure'
1690 006772      122      145      163  2$:     .ASCIZ  'Reserved'
1691 007003      102      165      163  3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
1692 007047      122      145      163  4$:     .ASCIZ  'Reserved'
1693                                    .EVEN
```

```
                                        .SBTTL  PRIPKT  - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
                                ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
                                ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
                                ;
                                ;INPUT:
                                ;       R0      NUMBER OF WORDS IN PACKET
                                ;       R3      HIGH ORDER COMMAND PACKET ADDRESS
                                ;       R4      ADDRESS OF COMMAND PACKET
                                ;       NOTE:   R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
007060                          PRIPKT::
007060                                  SAVREG                          ;SAVE THE REGISTERS
007064  010005                          MOV     R0,R5                   ;SAVE NO. OF WORDS IN PACKET
007066  005737  003102                  TST     KTENABLE                ;ABOVE 28K UNDER TEST?
007072  001001                          BNE     10$                     ;BR IF YES
007074  005003                          CLR     R3                      ;SET HIGH ORDER ADDRESS TO 0
007076  010301                  10$:    MOV     R3,R1                   ;COPY HIGH ORDER ADDRESS
007100  010400                          MOV     R4,R0                   ;GET LOWER ADDRESS
007102  006100                          ROL     R0                      ;SHIFT BIT 15 INTO C BIT
007104  006101                          ROL     R1                      ;AND INTO HIGH ORDER.
007106                                  PRINTB  #PKTADD,R1,R4           ;PRINT PACKET ADDRESS
007106  010446                          MOV     R4,-(SP)
007110  010146                          MOV     R1,-(SP)
007112  012746  007264                  MOV     #PKTADD,-(SP)
007116  012746  000003                  MOV     #3,-(SP)
007122  010600                          MOV     SP,R0
007124  104414                          TRAP    C$PNTB
007126  062706  000010                  ADD     #10,SP
007132  010300                  15$:    MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
007134  001404                          BEQ     20$                     ;BR IF NOT ABOVE 28K.
007136  010401                          MOV     R4,R1                   ;GET LOW ORDER ADDRESS
007140  004737  020252                  JSR     PC,SETMAP               ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
007144  010004                          MOV     R0,R4                   ;GET RETURNED PAR6 ADDRESS BIAS
007146  005001                  20$:    CLR     R1                      ;SAVE WORD NUMBER
007150  012402                  25$:    MOV     (R4)+,R2                ;GET PACKET CONTENTS
007152                                  PRINTB  #PKTFRM,R1,R2           ;PRINT THE DATA
007152  010246                          MOV     R2,-(SP)
007154  010146                          MOV     R1,-(SP)
007156  012746  007226                  MOV     #PKTFRM,-(SP)
007162  012746  000003                  MOV     #3,-(SP)
007166  010600                          MOV     SP,R0
007170  104414                          TRAP    C$PNTB
007172  062706  000010                  ADD     #10,SP
007176  005201                          INC     R1                      ;NEXT WORD NUMBER
007200  020105                          CMP     R1,R5                   ;DONE ALL PACKET WORDS?
007202  002762                          BLT     25$                     ;LOOP TILL ALL DONE
007204                                  PRINTB  #PKTNEW                 ;JUST A COUPLE NEW LINES
007204  012746  007321                  MOV     #PKTNEW,-(SP)
007210  012746  000001                  MOV     #1,-(SP)
007214  010600                          MOV     SP,R0
007216  104414                          TRAP    C$PNTB
007220  062706  000004                  ADD     #4,SP
007224  000207                          RTS     PC                      ;RETURN
007226    045   116   045  PKTFRM:      .ASCIZ  '%N%A Packet Word #%D1%A = %06'
007264    045   116   045  PKTADD:      .ASCIZ  '%N%A Packet Address = %01%05'
007321    045   116   045  PKTNEW:      .ASCIZ  '%N%N%A  '
                                        .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
                                ;
                                ;-

007332                          PRIBXOR::
007332                                  SAVREG                          ;SAVE THE REGISTERS
007336  010203                          MOV     R2,R3                   ;EXPECTED DATA
007340                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007350  012700  177400                  MOV     #^C<377>,R0             ;BYTE MASK
007354  040001                          BIC     R0,R1                   ;SAVE LOW BYTE RECV
007356  040002                          BIC     R0,R2                   ;SAVE LOW BYTE EXPD
007360  040003                          BIC     R0,R3                   ;SAVE LOW BYTE XOR
007362                                  PRINTB  #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
007362  010346                          MOV     R3,-(SP)
007364  010146                          MOV     R1,-(SP)
007366  010246                          MOV     R2,-(SP)
007370  012746  007414                  MOV     #XORBFOR,-(SP)
007374  012746  000004                  MOV     #4,-(SP)
007400  010600                          MOV     SP,R0
007402  104414                          TRAP    C$PNTB
007404  062706  000012                  ADD     #12,SP
007410  010300                          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007412  000207                          RTS     PC                      ;RETURN TO CALLER

007414      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
                                ;
                                ;-

007462                          PRIXOR::
007462                                  SAVREG                          ;SAVE THE REGISTERS
007466  010203                          MOV     R2,R3                   ;EXPECTED DATA
007470                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007500                                  PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
007500  010346                          MOV     R3,-(SP)
007502  010146                          MOV     R1,-(SP)
007504  010246                          MOV     R2,-(SP)
007506  012746  007532                  MOV     #XORFOR,-(SP)
007512  012746  000004                  MOV     #4,-(SP)
007516  010600                          MOV     SP,R0
007520  104414                          TRAP    C$PNTB
007522  062706  000012                  ADD     #12,SP
007526  010300                          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007530  000207                          RTS     PC                      ;RETURN TO CALLER

007532  045  116  045  XORFOR: .ASCIZ  '%N%A EXPD: %06%A RECV: %06%A XOR: %06'
                                .EVEN
```

```
                                        .SBTTL  PRIEQU  - PRINT BIT NUMBERS AS ASCII EQUIVALENT

                                ;+
                                ;
                                ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
                                ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
                                ;
                                ;INPUTS:
                                ;
                                ;       R0      OCTAL VALUE TO CONVERT
                                ;       R1      TABLE OF POINTERS TO ASCII EQUIVALENT
                                ;
                                ;-

007600                          PRIEQU:
007600                                  SAVREG                          ;SAVE THE REGISTERS
007604  000207                          RTS     PC                      ;RETURN TO CALLER


                                        .SBTTL  PRIRAM  - PRINT RAM ADDRESS
                                ;+
                                ;
                                ;PRINT CONTROLLER RAM ADDRESS.
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      RAM ADDRESS
                                ;
                                ;-
007606                          PRIRAM:
007606                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
007612                                  PRINTB  #RAMFOR,R4              ;PRINT RAM ADDRESS IN ERROR
007612  010446                          MOV     R4,-(SP)
007614  012746  007636                  MOV     #RAMFOR,-(SP)
007620  012746  000002                  MOV     #2,-(SP)
007624  010600                          MOV     SP,R0
007626  104414                          TRAP    C$PNTB
007630  062706  000006                  ADD     #6,SP
007634  000207                          RTS     PC                      ;RETURN

007636      045     116     045 RAMFOR: .ASCIZ  '%N%A CONTROLLER RAM ADDRESS = %O6'
                                        .EVEN


                                        .SBTTL  PRIADD  - PRINT MEMORY ERROR ADDRESS
                                ;+
                                ;
                                ;PRINT MEMORY ADDRESS
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ; IMPLICIT INPUTS
                                ;
                                ;       ERRHI   - HIGH ORDER ADDRESS
                                ;       ERRLO   - LOW ORDER ADDRESS
```

PRIADD - PRINT MEMORY ERROR ADDRESS

```
                                        ;
                                        ;-
                                        PRIADD:
007700                                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
007700  013700  002202                          MOV     ERRHI,R0                ;GET HIGH ADDRESSS
007704  013701  002204                          MOV     ERRLO,R1                ;GET LOW ADDRESS
007710  010102                                  MOV     R1,R2                   ;COPY LOW ADDRESS
007714  006101                                  ROL     R1                      ;SHIFT BIT 15 TO C BIT
007716  006100                                  ROL     R0                      ;SHIFT INTO HIGH ORDER
007720                                          PRINTB  #PRIA0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
007720  010246                                  MOV     R2,-(SP)
007722  010046                                  MOV     R0,-(SP)
007724  012746  007750                          MOV     #PRIA0,-(SP)
007730  012746  000003                          MOV     #3,-(SP)
007734  010600                                  MOV     SP,R0
007736  104414                                  TRAP    C$PNTB
007740  062706  000010                          ADD     #10,SP
007744  000207                                  RTS     PC                      ;RETURN

007750    045    116    045  PRIA0:  .ASCIZ  '%N%A MEMORY ERROR ADDRESS = %01%05'
                                   .EVEN
                                        ;+
                                        ;PRINT MEMORY ADDRESS
                                        ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                        ; IMPLICIT INPUTS
                                        ;       ERRHI   - HIGH ORDER ADDRESS
                                        ;       ERRLO   - LOW ORDER ADDRESS
                                        ;-
010014                                  PRITADD:
010014                                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010020  013700  002202                          MOV     ERRHI,R0                ;GET HIGH ADDRESSS
010024  013701  002204                          MOV     ERRLO,R1                ;GET LOW ADDRESS
010030  010102                                  MOV     R1,R2                   ;COPY LOW ADDRESS
010032  006101                                  ROL     R1                      ;SHIFT BIT 15 TO C BIT
010034  006100                                  ROL     R0                      ;SHIFT INTO HIGH ORDER
010036                                          PRINTB  #PRIT0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
010036  010246                                  MOV     R2,-(SP)
010040  010046                                  MOV     R0,-(SP)
010042  012746  010064                          MOV     #PRIT0,-(SP)
010046  012746  000003                          MOV     #3,-(SP)
010052  010600                                  MOV     SP,R0
010054  104414                                  TRAP    C$PNTB
010056  062706  000010                          ADD     #10,SP
010062  000207                                  RTS     PC                      ;RETURN
010064    045    116    045  PRIT0:  .ASCIZ  '%N%A MEMORY TEST ADDRESS = %01%05'
                                   .EVEN
```

```
                                        .SBTTL  SPACE   - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
                                ;+
                                ;ROUTINE TO ISSUE A SPACE RECORDS
                                ;COMMAND (FORWARD OR REVERSE)
                                ;INPUT:
                                ;       R3      NUMBER OF RECORDS TO BE SPACED OVER
                                ;               BIT15 CONTROLS DIRECTION
                                ;               BIT15 = 0 IS FORWARD
                                ;               BIT15 = 1 IS REVERSE
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
                                ;OUTPUT:
                                ;       CARRY   SET - SPACE RECORDS COMMAND OK
                                ;               CLR - SPACE RECORDS FAILED
                                ;       R0      THE CONTENTS OF R4 IS MOVED TO R0
                                ;IMPLICIT OUTPUT:
                                ;       TAPE HAS BEEN MOVED
                                ;SIDE EFFECTS:
                                ;-
010126                          SPACE::
010126                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
010132  012737  000764  010320          MOV     #500.,SDELAY            ;SET UP DELAY
010140  012737  140010  010310          MOV     #140010,80$             ;SET UP COMMAND, SPACE FORWARD
010146  005703                          TST     R3                      ;CHECK FOR DIRECTION
010150  100403                          BMI     5$                      ;BR, IF REVERSE INDICATED
010152  010337  010312                  MOV     R3,90$                  ;LOAD UP NUMBER OF RECORDS TO SPACE
010156  000407                          BR      10$                     ;GO DO COMMAND
010160  042703  100000          5$:     BIC     #BIT15,R3               ;CLEAR DIRECTION BIT
010164  010337  010312                  MOV     R3,90$                  ;LOAD UP NUMBER OF RECORDS TO SPACE
010170  052737  000400  010310          BIS     #BIT8,80$               ;SET REVERSE BIT IN COMMAND PACKET
010176  012704  010310          10$:    MOV     #80$,R4                 ;SET UP R4 WITH PACKET ADDRESS
010202  010465  177776                  MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010206  004737  017104          15$:    JSR     PC,WAITF                ;WAIT FOR SSR
010212  103420                          BCS     20$                     ;BR, IF SSR IS SET AND OK
010214                                  DELAY   250                     ;DELAY ABOUT .25 SECONDS
010214  012727  000250                  MOV     #250.,(PC)+
010220  000000                          .WORD   0
010222  013727  002116                  MOV     L$DLY,(PC)+
010226  000000                          .WORD   0
010230  005367  177772                  DEC     -6(PC)
010234  001375                          BNE     .-4
010236  005367  177756                  DEC     -22(PC)
010242  001367                          BNE     .-20
010244  005337  010320                  DEC     $DELAY                  ;BUMP DELAY COUNTER DOWN
010250  001356                          BNE     15$                     ;BR, IF MORE DELAY
010252  000411                          BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
010254  016501  000000          20$:    MOV     TSSR(R5),R1             ;READ TSSR
010260  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
010264  020201          25$:    CMP     R2,R1                   ;ARE THEY OK
010266  001401                          BEQ     40$                     ;BR, IF EQUAL = OK
010270  000402                          BR      60$                     ;TROUBLE EXIT
010272  000261          40$:    SEC                             ;SET CARRY NO TROUBLE
010274  000401                          BR      70$                     ;EXIT
010276  000241          60$:    CLC                             ;CARRY CLEAR = ERROR
010300                  70$:
010300  010400                          MOV     R4,R0                   ;PASS PACKET ADDRESS
010302  000207                          RTS     PC                      ;RETURN
```

```
                                        ;
                                        ;PACKET FOR SPACE COMMAND
                                        ;
010304                                          .BLKB   10-<.-TUV2A&7>
                                        ;
                                        ;COMMAND WORD
010310  000000                          80$:    .WORD
                                        ;NUMBER OF RECORDS TO BE SPACED OVER WORD
010312  000000                          90$:    .WORD
010314  000000                                  .WORD
010316  000000                                  .WORD
010320  000000                          SDELAY: .WORD   0               ;DELAY COUNTER
                                                .EVEN
```

```
                        .SBTTL  WRTCHR  - WRITE CHARACTERISTICS COMMAND

                ;+
                ;
                ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
                ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
                ;
                ;INPUT:
                ;
                ;       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 FOLLOWS:
                ;               BENBSW = BUFFER ENABLE SWITCH ON OR OFF
                ;
                ;
                ;SIDE EFFECTS:
                ;
                ;
                ;-

010322                  WRTCHR::
010322                          SAVREG                  ;SAVE THE GENERAL REGISTERS
010326  005037  002174          CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010332  010465  177776  10$:    MOV     R4,TSDB(R5)     ;SEND OUT COMMAND
010336  004737  017220          JSR     PC,CHKTSSR      ;WAIT FOR SSR
010342  103401                  BCS     20$             ;BR, IF SSR IS SET AND OK
010344  000423                  BR      60$             ;BR IF TROUBLE CARRY = CLEAR
010346  016501  000000  20$:    MOV     TSSR(R5),R1     ;READ TSSR
010352  012702  000200          MOV     #SSR,R2         ;SET UP EXPECTED
010356  032701  000100          BIT     #OFL,R1         ;WAS OFF LINE SET IN TSSR
010362  001402                  BEQ     25$             ;BR, IF NO OFL SET
010364  052702  000100          BIS     #OFL,R2         ;MAKE THEM LOOK ALIKE
010370  020201          25$:    CMP     R2,R1           ;ARE THEY OK
010372  001401                  BEQ     40$             ;BR, IF EQUAL = OK
010374  000407                  BR      60$             ;TROUBLE EXIT
010376  062704  000010  40$:    ADD     #8.,R4          ;POINT TO WRT CHARA DATA PACKET
010402  011403                  MOV     (R4),R3         ;GET ADDRESS OF MESSAGE BUFFER
010404  010337  002716          MOV     R3,MESBFA       ;STORE FOR PRINT ROUTINES
010410  000261                  SEC                     ;SET CARRY NO TROUBLE
010412  000401                  BR      70$             ;EXIT
010414  000241          60$:    CLC                     ;CARRY CLEAR = ERROR
010416  016500  000000  70$:    MOV     TSSR(R5),R0     ;RETURN TSSR CONTENTS
010422  000207                  RTS     PC              ;RETURN
```

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

                                ;+
                                ;
                                ;THIS ROUTINE WILL REWIND THE SELECTED TAPE.
                                ;
                                ;       CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
                                ;                TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
                                ;                SSR TO SET IN THE TSSR
                                ;
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       DO A SOFT INIT
                                ;       DO A WRITE CHARACTERISTICS
                                ;       JSR     PC,REWIND
                                ;
                                ;INPUT:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;
                                ;OUTPUT
                                ;
                                ;       R0      THE CONTENTS OF R4 IS PASSED TO R0
                                ;
                                ;
                                ;-
010424                          REWIND::
010424                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
010430  012704  010520                  MOV     #RWPACK,R4      ;GET PACKET ADDRESS
010434  010465  177776                  MOV     R4,TSDB(R5)     ;SEND PACKET ADDRESS TO EXECUTE
010440  012703  000550                  MOV     #360.,R3        ;ENOUGH TIME FOR 2400' REEL TO REWIND
010444  004737  017104          10$:    JSR     PC,WAITF        ;WAIT FOR SSR TO SET
010450  103417                          BCS     20$             ;LEAVE WHEN SSR IS SET
010452                                  DELAY   250.            ;WAIT FOR .25 SECONDS
010452  012727  000372                  MOV     #250.,(PC)+
010456  000000                          .WORD   0
010460  013727  002116                  MOV     L$DLY,(PC)+
010464  000000                          .WORD   0
010466  005367  177772                  DEC     -6(PC)
010472  001375                          BNE     .-4
010474  005367  177756                  DEC     -22(PC)
010500  001367                          BNE     .-20
010502  005303                          DEC     R3              ;BUMP COUNTER DOWN
010504  001357                          BNE     10$             ;KEEP GOING
010506  000241                          CLC                     ;CLEAR CARRY TO SET ERROR
010510  010400                  20$:    MOV     R4,R0           ;PASS THE PACKET ADDRESS
010512  000207                          RTS     PC              ;RETURN
010514                                  .BLKB   10-<.-TUV2A&7>
010520                          RWPACK:
010520  102010                          .WORD   102010          ;POSTION COMMAND (REWIND)
010522  000000                          .WORD   0               ;NOT USED
```

```
                        .SBTTL  CKRAM   - COMPARE RAM TO I/O PACKET

                ;+
                ;
                ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
                ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
                ;
                ;INPUT:
                ;
                ;       R4      ADDRESS OF THE COMMAND PACKET
                ;       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. FOR PRAMPKT ROUTINE
                ;
                ;SIDE EFFECTS:
                ;
                ;
                ;-

010524                  CKRAM::
010524                          SAVREG                  ;SAVE THE GENERAL REGISTERS
010530  012701  002206          MOV     #RAMDATA,R1     ;ADDRESS TO SAVE THE RAM DATA
010534  012702  000020          MOV     #RMPKTBEG,R2    ;BYTE ADDRESS OF FIRST RAM DATA
010540  005003                  CLR     R3              ;CLEAR THE ERROR FLAG
010542  004737  017220          JSR     PC,CHKTSSR      ;WAIT FOR SSR
010546  004737  017220  10$:    JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
010552  110265  177777          MOVB    R2,TSDBH(R5)    ;SELECT NEXT RAM ADDRESS
010556  004737  017220          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
010562  116511  177776          MOVB    TSBAL(R5),(R1)  ;READ THE RAM DATA
010566  122124                  CMPB    (R1)+,(R4)+     ;COMPARE TO EXPECTED
010570  001401                  BEQ     20$             ;BRANCH IF OK
010572  005203                  INC     R3              ;SET ERROR FLAG
010574  005202          20$:    INC     R2              ;ADDRESS OF NEXT RAM LOCATION
010576  020227  000027          CMP     R2,#RMPKTEND    ;REACHED END YET ?
010602  003761                  BLE     10$             ;BRANCH TILL ALL READ
010604  005703                  TST     R3              ;WAS AN ERROR FOUND ?
010606  001402                  BEQ     30$             ;BRANCH IF NOT
010610  000241                  CLC                     ;CLEAR CARRY TO SHOW ERROR
010612  000401                  BR      50$             ;AND EXIT
010614  000261          30$:    SEC                     ;SHOW GOOD COMPARE
010616  012737  000010  002246  50$:    MOV     #8.,RAMSIZ      ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
010624  000207                  RTS     PC              ;RETURN
```

```
                                        .SBTTL RAMER - READ AND DISPLAY SELECTED RAM
                                ;+
                                ;
                                ;ROUTINE TO READ THE SELECTED RAM LOCATIONS
                                ;
                                ;INPUT:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       CONSOLE WILL ALSO BE PRINTED TO
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       THE TABLE RAMDATA IS FILLED WITH THE
                                ;       DATA HELD IN RAM.
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;
                                ;-

                                RAMER::
010626                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
010626                                  MOV     RAMR5H,R5               ;RESET R5 TO FIRST DEVICE REGISTER
010632  013705  011012                    MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
010636  012701  002206                    MOV     RAMHLD,R2               ;BYTE ADDRESS OF THE FIRST RAM DATA
010642  013702  011010                    MOV     RAMSIZ,R3               ;SET THE SIZE OF THE READ UP
010646  013703  002246            10$:    JSR     PC,CHKTSSR              ;WAIT FOR THE SSR TO SET
010652  004737  017220                    MOVB    R2,TSDBH(R5)            ;SELECT NEXT RAM ADDRESS
010656  110265  177777                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
010662  004737  017220                    MOVB    TSBAL(R5),(R1)+         ;READ THE RAM DATA
010666  116521  177776            20$:    ADD     #1,R2                   ;ADDRESS OF THE NEXT RAM LOCATION
010672  062702  000001                    SOB     R3,10$                  ;NUMBER OF LOCATIONS COUNTER
010676  077313                           MOV     RAMSIZ,R4               ;GET THE RAM SIZE
010700  013704  002246                    MOV     RAMHLD,R2               ;GET THE STARTING RAM ADDRESS
010704  013702  011010                    ADD     R2,R4                   ;CALCULATE THE END ADDRESS
010710  060204                           SUB     #1,R4                   ;CORRECT VALUE OF PRINTOUT
010712  162704  000001                    PRINTX  #RAMIOP,R2,R4           ;RAM ADDRESS = 10 - 17, ETC.
010716                                   MOV     R4,-(SP)
010716  010446                           MOV     R2,-(SP)
010720  010246                           MOV     #RAMIOP,-(SP)
010722  012746  011014                    MOV     #3,-(SP)
010726  012746  000003                    MOV     SP,R0
010732  010600                           TRAP    C$PNTX
010734  104415                           ADD     #10,SP
010736  062706  000010                    MOV     #RAMDATA,R1             ;ADDRESS OF WHERE RAM DATA IS
010742  012701  002206                    MOV     RAMSIZ,R3               ;THE SIZE OF THE RAM FIELD READ
010746  013703  002246            30$:    CLR     R4                      ;NO EXTRA DATA LEFT OVER
010752  005004                           MOVB    (R1)+,R4                ;PICK UP BYTE OF RAM DATA
010754  112104                           BIC     #177400,R4              ;GET RID OF SIGN EXTEND
010756  042704  177400                    PRINTX  #RAMPD,R4               ;"010 211 111 222 377 000 123 134 ETC."
010762                                   MOV     R4,-(SP)
010762  010446                           MOV     #RAMPD,-(SP)
010764  012746  011065                    MJV     #2,-(SP)
010770  012746  000002                    MOV     SP,R0
010774  010600                           TRAP    C$PNTX
010776  104415                           ADD     #6,SP
011000  062706  000006                    SOB     R3,30$                  ;LOOP UNTIL ALL PRINTED
011004  077316
```

```
 2156 011006  000207                  50$:    RTS     PC              ;RETURN
 2157
 2158 011010  000000                  RAMHLD: .WORD   0               ;RAM ADDR HOLDER 1ST ADDRESS
 2159 011012  000000                  RAMR5H: .WORD   0               ;HOLDS R5 FOR LATER
 2160 011014     045     116     045  RAMIOP: .ASCIZ  '%N%A Ram Address (Octal) = %03%A - %03%N'
 2161 011065     045     101     040  RAMPD:  .ASCIZ  '%A %03%A '
 2162                                          .EVEN
 2163
```

```
                                        .SBTTL  CKRAM2  - COMPARE RAM TO I/O CHARACTERISTICS DATA
                                ;+
                                ;
                                ;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:
                                ;
                                ;
                                ;-

011100                          CKRAM2::
011100                                  SAVREG                  ;SAVE THE GENERAL REGISTERS
011104  012701  002206                  MOV     #RAMDATA,R1     ;ADDRESS TO SAVE THE RAM DATA
011110  012702  000167                  MOV     #RMCHBEG,R2     ;BYTE ADDRESS OF FIRST RAM DATA
011114  005003                          CLR     R3              ;CLEAR THE ERROR FLAG
011116  004737  017220                  JSR     PC,CHKTSSR      ;WAIT FOR SSR
011122  004737  017220          10$:    JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
011126  110265  177777                  MOVB    R2,TSDBH(R5)    ;SELECT NEXT RAM ADDRESS
011132  004737  017220                  JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
011136  116511  177776                  MOVB    TSBAL(R5),(R1)  ;READ THE RAM DATA
011142  122124                          CMPB    (R1)+,(R4)+     ;COMPARE TO EXPECTED
011144  001401                          BEQ     20$             ;BRANCH IF OK
011146  005203                          INC     R3              ;SET ERROR FLAG
011150  005202                  20$:    INC     R2              ;ADDRESS OF NEXT RAM LOCATION
011152  012737  000010  002246          MOV     #8.,RAMSIZ      ;ASSUME NORMAL NOT SET
011160  020227  000176                  CMP     R2,#RMCHEND-2   ;REACHED END YET ?
011164  003756                          BLE     10$             ;BRANCH TILL ALL READ
011166  005703                  27$:    TST     R3              ;WAS AN ERROR FOUND ?
011170  001402                          BEQ     30$             ;BRANCH IF NOT
011172  000241                          CLC                     ;CLEAR CARRY TO SHOW ERROR
011174  000401                          BR      50$             ;AND EXIT
011176  000261                  30$:    SEC                     ;SHOW GOOD COMPARE
011200  000207                  50$:    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 TO RECV DATA
                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                                ;
                                ;-
                                CKMSG::
                                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
                                        MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
                                        MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
                                        TST     KTENABLE                ;TESTING ABOVE 28K?
                                        BEQ     10$                     ;BR IF NO
                                        JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
                                        MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
                                10$:    CLR     R4                      ;WORD IN BUFFER
                                        CLR     R3                      ;CLEAR ERROR SEEN FLAG
                                        MOV     R2,R5                   ;GET EXPD BUFFER ADDRESS
                                15$:    MOV     (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
                                        MOV     (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
                                        CMP     (R2)+,(R1)+             ;EXPD EQUAL RECV?
                                        BEQ     25$                     ;BR IF YES
                                        INC     R3                      ;SET ERROR SEEN FLAG
                                25$:    ADD     #2,R4                   ;POINT TO NEXT WORD ADDRESS
                                        CMP     R4,#14                  ;DONE FIRST 7 WORDS?
                                        BLE     15$                     ;BR IF NO
                                        BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
                                        BEQ     50$                     ;BR IF NO
                                        CMP     R4,#16                  ;DONE EXTENDED FEATURES WORD?
                                        BLE     15$                     ;BR IF NO
                                50$:    TST     R3                      ;ANY ERRORS SEEN?
                                        BEQ     55$                     ;BR IF NO
                                        CLC                             ;SET FAILURE
                                        BR      60$                     ;
                                55$:    SEC                             ;SET SUCCESS
                                60$:    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
                                ;
                                ;-
                                CKMSG2::
                                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
                                        CMP     R3,#RECMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
                                        BLE     5$                      ;@@D BR IF NO
                                        MOV     #RECMSG-EXPMSG,R3;@@D
                                        PRINTF  #DEBUGMSG               ;@@D
                                        MOV     #DEBUGMSG,-(SP)
                                        MOV     #1,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTF
                                        ADD     #4,SP
                                5$:     MOV     R0,RCVHIADD     ;SAVE RECV HIGH ADDRESS
                                        MOV     R1,RCVLOAD      ;SAVE RECV LOW ADDRESS
                                        TST     KTENABLE        ;TESTING ABOVE 28K?
                                        BEQ     10$             ;BR IF NO
                                        JSR     PC,SETMAP       ;RETURN ADDRESS BIASED TO PAR6 IN R0
                                        MOV     R0,R1           ;GET RETURNED ADDRESS BIASED TO PAR6
                                10$:    CLR     R4              ;WORD IN BUFFER
                                        CLR     R5              ;CLEAR ERROR SEEN FLAG
                                15$:    MOVB    (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
                                        MOVB    (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
                                        CMPB    (R2)+,(R1)+     ;EXPD EQUAL RECV?
                                        BEQ     25$             ;BR IF YES
                                        INC     R5              ;SET ERROR SEEN FLAG
                                25$:    ADD     #1,R4           ;POINT TO NEXT BYTE
                                        CMP     R4,R3           ;DONE ALL BYTES?
                                        BGE     50$             ;BR IF YES
                                        BR      15$             ;DO NEXT BYTE
                                50$:    TST     R5              ;ANY ERRORS SEEN?
                                        BEQ     55$             ;BR IF NO
```

```
2325 011444  000241                        CLC                     ;SET FAILURE
2326 011446  000401                        BR      60$             ;
2327 011450  000261                55$:    SEC                     ;SET SUCCESS
2328 011452  000207                60$:    RTS     PC              ;RETURN
2329
2330 011454     120     122     117  DEBUGMSG:      .ASCIZ  'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';aad
2331 011544     045     116     045  FERCM:  .ASCII  /%N%A  ***/
2332 011555     040     040     124  ERCM:   .ASCIZ  /  TSSR ERROR CODE REC'D = /
2333 011610     056     056     056  SIMSG:  .ASCIZ  /.... AFTER DOING SOFT INIT/
2334 011643     124     105     123  TINERR: .ASCIZ  /TEST: .../
2335                                   .EVEN
```

```
                                        ;+
                                        ;
                                        ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
                                        ;
                                        ;INPUT:
                                        ;
                                        ;       R1      CONTENTS OF TSSR AT ERROR
                                        ;
                                        ;SIDE EFFECTS:
                                        ;
                                        ;       EXECUTES DROP UNIT TO CEASE TESTING
                                        ;
                                        ;-

011656                                          BGNMSG  SFIMSG
011656                                  SFIMSG::
011656  004737  005264                          JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
011662  004737  020136                          JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
011666                                          ENDMSG
011666                                  L10003:
011666  104423                                  TRAP    C$MSG

                                        ;+
                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                        ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;
                                        ;-

011670                                          BGNMSG  PKTSSR
011670                                  PKTSSR::
011670  004737  005264                          JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
011674  012700  000004                          MOV     #4,R0           ;NO. OF WORDS IN PACKET
011700  004737  007060                          JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
011704  013700  002716                          MOV     MESBFA,R0       ;ADDRESS OF MESSAGE BUFFER
011710  005001                                  CLR     R1              ;ASSUME NO HIGH MEMORY
011712  004737  014052                          JSR     PC,PRMESS       ;PRINT THE MESSAGE BUFFER ALSO
011716                                          ENDMSG
011716                                  L10004:
011716  104423                                  TRAP    C$MSG

                                        ;+
                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                        ;TSSR AND A GET STATUS COMMAND PACKET.
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;
                                        ;-
```

```
                                        BGNMSG  PKTGETS
                                PKTGETS::
011720  004737  005264                  JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
011724  012700  000002                  MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
011730  004737  007060                  JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
011734                                  ENDMSG
                                L10005:
011734  104423                          TRAP    C$MSG


                                ;+
                                ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      TSSR CONTENTS
                                ;       R4      ADDRESS OF COMMAND PACKET
                                ;-

                                        BGNMSG  SFFMSG
                                SFFMSG::
011736  004737  005264                  JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
011742                                  ENDMSG
                                L10006:
011742  104423                          TRAP    C$MSG


                                        .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
                                ;BUFFER FOR ERROR REPORTS
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      CONTENTS OF TSSR
                                ;       R2      LOW ORDER MESSAGE BUFFER
                                ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
                                ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                ;-
                                        BGNMSG  PKTMES
                                PKTMES::
011744  004737  005264                  JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR
011750  010200                          MOV     R2,R0           ;LOW ORDER ADDRESS
011752  010301                          MOV     R3,R1           ;HIGH ORDER ADDRESS
011754  004737  014052                  JSR     PC,PRMESS       ;PRINT THE MESSAGE BUFFER
011760                                  ENDMSG
                                L10007:
011760  104423                          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
                                ;-
                                
011762                                  BGNMSG  ADDSSR
011762                          ADDSSR::
011762  004737  010014                  JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
011766  016501  000000                  MOV     TSSR(R5),R1     ;GET CURRENT TSSR
011772  004737  005264                  JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
011776                                  ENDMSG
011776                          L10010:
011776  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
                                ;-
                                
012000                                  BGNMSG  MSGEXP
012000                          MSGEXP::
012000  012700  000007                  MOV     #7,R0           ;ASSUME NO EXT FEATURES
012004  004737  015406          5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012010                                  ENDMSG
012010                          L10011:
012010  104423                          TRAP    C$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
                                        ;       RECMSG  - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
                                        ;-
                                                BGNMSG  FIFEXP
012012                                  FIFEXP::
                                                PRINTX  #FIF1MSG,R1     ;PRINT BYTES TRANSFERRED
012012  010146                                  MOV     R1,-(SP)
012014  012746  012064                          MOV     #FIF1MSG,-(SP)
012020  012746  000002                          MOV     #2,-(SP)
012024  010600                                  MOV     SP,R0
012026  104415                                  TRAP    C$PNTX
012030  062706  000006                          ADD     #6,SP
012034                                          PRINTX  #FIF2MSG        ;PRINT HEADER MSG
012034  012746  012133                          MOV     #FIF2MSG,-(SP)
012040  012746  000001                          MOV     #1,-(SP)
012044  010600                                  MOV     SP,R0
012046  104415                                  TRAP    C$PNTX
012050  062706  000004                          ADD     #4,SP
012054  010100                                  MOV     R1,R0           ;GET BYTE COUNT
012056  004737  015756                          JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
012062                                          ENDMSG
012062                                  L10012:
012062  104423                                  TRAP    C$MSG
012064    045   116   045   FIF1MSG:            .ASCIZ  '%N%A NUMBER OF BYTES TRANSFERRED = %D2'
012133    045   116   045   FIF2MSG:            .ASCIZ  '%N%A FIFO DATA BYTES IN ERROR:'
                                                .EVEN
```

```
                                        .SBTTL  MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
012172                          BGNMSG  MSGSTAT
012172                  MSGSTAT::
012172  012701  012234          MOV     #STATCOD,R1     ;ASCII ADDRESS TABLE
012176  012100          10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
012200  001410                  BEQ     20$             ;BR IF YES
012202                          PRINTX  R0              ;PRINT STATUS BIT NAMES
012202  010046                  MOV     R0,-(SP)
012204  012746  000001          MOV     #1,-(SP)
012210  010600                  MOV     SP,R0
012212  104415                  TRAP    C$PNTX
012214  062706  000004          ADD     #4,SP
012220  000766                  BR      10$             ;DO ANOTHER MSG LINE
012222  012700  000012  20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
012226  004737  015406          JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012232                          ENDMSG
012232                  L10013:
012232  104423                  TRAP    C$MSG

012234  012252  012314  012405  STATCOD:        .WORD   1$,2$,3$,4$,5$,6$,0
012252    045     116     045   1$:.ASCIZ  '%N%A Tape Bus Signals in Word #8:'
012314    045     116     045   2$:.ASCIZ  '%N%A       PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>'
012405    045     116     045   3$:.ASCIZ  '%N%A       IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>'
012476    045     116     045   4$:.ASCIZ  '%N%A       IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>'
012567    045     116     045   5$:.ASCIZ  '%N%A Tape Bus Signals in Word #9:'
012631    045     116     045   6$:.ASCIZ  '%N%A       DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>'
                                .EVEN


                                        .SBTTL  MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
012706                          BGNMSG  MSGLOOP
012706                  MSGLOOP::
012706  012701  012750          MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
```

MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

```
 2540 012712  012100                10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
 2541 012714  001410                        BEQ     20$             ;BR IF YES
 2542 012716                                PRINTX  R0              ;PRINT STATUS BIT NAMES
      012716  010046                        MOV     R0,-(SP)
      012720  012746  000001                MOV     #1,-(SP)
      012724  010600                        MOV     SP,R0
      012726  104415                        TRAP    C$PNTX
      012730  062706  000004                ADD     #4,SP
 2543 012734  000766                        BR      10$             ;DO ANOTHER MSG LINE
 2544 012736  012700  000012        20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
 2545 012742  004737  015406                JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
 2546 012746                                ENDMSG
      012746                        L10014:
      012746  104423                        TRAP    C$MSG
 2547
 2548 012750  012770  013043  013142  LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
 2549 012770    045     116     045  1$:.ASCIZ  '%N%A Tape Bus Loopback Signals in Word #8:'
 2550 013043    045     116     045  2$:.ASCIZ  '%N%A      PARERR<15>          IRESV2<14>          IRESV1<13>'
 2551 013142    045     116     045  3$:.ASCIZ  '%N%A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
 2552 013241    045     116     045  4$:.ASCIZ  '%N%A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
 2553 013340    045     116     045  5$:.ASCIZ  '%N%A ITAD0=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
 2554 013437    045     116     045  6$:.ASCIZ  '%N%A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
 2555 013536    045     116     045  7$:.ASCIZ  '%N%A IGO  =>IFPT<00>'
 2556                                        .EVEN
 2557
```

```
                                        .SBTTL  MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
013564                                  BGNMSG  MSGSUB
013564                          MSGSUB::
013564  012700  000012                  MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
013570  004737  015406                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013574                                  ENDMSG
013574                          L10015:
013574  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
                                ;-
013576                                  BGNMSG  MEMADD
013576                          MEMADD::
013576  004737  007700                  JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
013602  013701  002176                  MOV     EXPD,R1         ;GET EXPD DATA
013606  013702  002200                  MOV     RECV,R2         ;GET RECEIVED DATA
013612  004737  007462                  JSR     PC,PRIXOR       ;PRINT EXPD/RECV
013616                                  ENDMSG
013616                          L10016:
013616  104423                          TRAP    C$MSG
```

PRAMPKT - PRINT RAM AND PACKET DATA

```
                                        .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
                                ;-

013620                          PRAMPKT:
013620                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
013624  012701  002206                  MOV     #RAMDATA,R1     ;DATA FROM THE RAM
013630  005002                          CLR     R2              ;INIT BYTE NUMBER
013632  122124                  5$:     CMPB    (R1)+,(R4)+     ;COMPARE EXPECTED, RECEIVED
013634  001000                          BNE     7$              ;BR IF NO MATCH
013636  116105  177777          7$:     MOVB    -1(R1),R5       ;GET RECV RAM DATA
013642  116403  177777                  MOVB    -1(R4),R3       ;GET EXPD PACKET DATA
013646                                  XOR     R5,R3           ;XOR EXPD/RECV
013656  042703  177400                  BIC     #177400,R3      ;LOW BYTE ONLY
013662  116137  177777  002200          MOVB    -1(R1),RECV     ;GET RECEIVED RAM DATA
013670  116437  177777  002176          MOVB    -1(R4),EXPD     ;GET EXPECTED RAM DATA
013676                                  PRINTB  #RAMASC,R2,RECV,EXPD,R3
013676  010346                          MOV     R3,-(SP)
013700  013746  002176                  MOV     EXPD,-(SP)
013704  013746  002200                  MOV     RECV,-(SP)
013710  010246                          MOV     R2,-(SP)
013712  012746  013766                  MOV     #RAMASC,-(SP)
013716  012746  000005                  MOV     #5,-(SP)
013722  010600                          MOV     SP,R0
013724  104414                          TRAP    C$PNTB
013726  062706  000014                  ADD     #14,SP
013732  005202                  10$:    INC     R2              ;UPDATE BYTE COUNT
013734  005737  002246                  TST     RAMSIZ          ;DEFAULT TO 8.?
013740  001404                          BEQ     15$             ;BR IF YES
013742  020237  002246                  CMP     R2,RAMSIZ       ;DONE ALL BYTES?
013746  003731                          BLE     5$              ;BR IF NO
013750  000403                          BR      25$             ;
013752  020227  000010          15$:    CMP     R2,#8.          ;DONE DEFAULT NUMBER OF BYTES?
013756  002725                  20$:    BLT     5$              ;BR IF NO
013760  005037  002246          25$:    CLR     RAMSIZ          ;SET DEFAULT RAMSIZ
013764  000207                          RTS     PC              ;RETURN

013766  045  116  045  RAMASC: .ASCIZ  '%N%A  BYTE: %D2%A RAM: %O3%A Packet: %O3%A XOR:%O3'
                                .EVEN
```

```
                                        .SBTTL  PRMESS  - PRINT CONTENTS OF MESSAGE BUFFER
                                ;+
                                ;
                                ;THIS ROUTINE PRINTS THE CONTENTS OF
                                ;THE 7 WORD MESSAGE BUFFER RETURNED BY THE
                                ;TU80.
                                ;
                                ;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  010537  011012                  MOV     R5,RAMR5H       ;SAVE DEVICE REGISTER POINTER
014062  010005                          MOV     R0,R5           ;SAVE LOW ORDER ADDRESS
014064  005737  003102                  TST     KTENABLE        ;ADDRESS ABOVE 28K?
014070  001001                          BNE     10$             ;BR IF YES
014072  005001                          CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
014074  010103                  10$:    MOV     R1,R3           ;SAVE HIGH ORDER ADDRESS
014076  006100                          ROL     R0              ;SHIFT BIT15 TO C BIT
014100  006101                          ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
014102                                  PRINTX  #PROASC,R1,R5   ;PRINT MESSAGE BUFFER ADDRESS
014102  010546                          MOV     R5,-(SP)
014104  010146                          MOV     R1,-(SP)
014106  012746  014677                  MOV     #PROASC,-(SP)
014112  012746  000003                  MOV     #3,-(SP)
014116  010600                          MOV     SP,R0
014120  104415                          TRAP    C$PNTX
014122  062706  000010                  ADD     #10,SP
014126  022715  177777                  CMP     #177777,(R5)    ;MESSAGE BUFFER FULL OF ONES
014132  001010                          BNE     15$             ;BR IF BUFFER IS PROBABLY OKAY
014134                                  PRINTX  #MESBFN         ;"MESSAGE BUFFER PROBABLY NOT VALID"
014134  012746  014617                  MOV     #MESBFN,-(SP)
014140  012746  000001                  MOV     #1,-(SP)
014144  010600                          MOV     SP,R0
014146  104415                          TRAP    C$PNTX
014150  062706  000004                  ADD     #4,SP
014154                          15$:    PRINTX  #PR1ASC         ;PRINT HEADER FOR CONTENTS
014154  012746  014744                  MOV     #PR1ASC,-(SP)
014160  012746  000001                  MOV     #1,-(SP)
014164  010600                          MOV     SP,R0
014166  104415                          TRAP    C$PNTX
014170  062706  000004                  ADD     #4,SP
014174  005004                          CLR     R4              ;NUMBER OF THE NEXT WORD
014176  010501                          MOV     R5,R1           ;COPY LOW ORDER ADDRESS
014200  010300                          MOV     R3,R0           ;COPY HIGH ORDER ADDRESS
014202  001403                          BEQ     20$             ;BR IF NOT ABOVE 28K
014204  004737  020252                  JSR     PC,SETMAP       ;SETUP PAR ADDRESS IN R0
014210  010005                          MOV     R0,R5           ;GET PAR FORMAT ADDRESS ABOVE 28K
014212                          20$:
014212                                  PRINTX  #MESHEA,(R5)+   ;PRINT "MESSAGE BUFFER HEADER ="
```

```
        014212  012546                          MOV     (R5)+,-(SP)
        014214  012746  015002                  MOV     #MESHEA,-(SP)
        014220  012746  000002                  MOV     #2,-(SP)
        014224  010600                          MOV     SP,R0
        014226  104415                          TRAP    C$PNTX
        014230  062706  000006                  ADD     #6,SP
2689    014234                                  PRINTX  #DATAFL,(R5)+   ;PRINT "DATA FIELD LENGTH     ="
        014234  012546                          MOV     (R5)+,-(SP)
        014236  012746  015047                  MOV     #DATAFL,-(SP)
        014242  012746  000002                  MOV     #2,-(SP)
        014246  010600                          MOV     SP,R0
        014250  104415                          TRAP    C$PNTX
        014252  062706  000006                  ADD     #6,SP
2690    014256                                  PRINTX  #RBPCRA,(R5)+   ;PRINT "RESIDUAL BYTE COUNTER ="
        014256  012546                          MOV     (R5)+,-(SP)
        014260  012746  015114                  MOV     #RBPCRA,-(SP)
        014264  012746  000002                  MOV     #2,-(SP)
        014270  010600                          MOV     SP,R0
        014272  104415                          TRAP    C$PNTX
        014274  062706  000006                  ADD     #6,SP
2691    014300                                  PRINTX  #XSOCON,(R5)+   ;PRINT "XSTAT0 CONTENTS       ="
        014300  012546                          MOV     (R5)+,-(SP)
        014302  012746  015161                  MOV     #XSOCON,-(SP)
        014306  012746  000002                  MOV     #2,-(SP)
        014312  010600                          MOV     SP,R0
        014314  104415                          TRAP    C$PNTX
        014316  062706  000006                  ADD     #6,SP
2692    014322                                  PRINTX  #XS1CON,(R5)+   ;PRINT "XSTAT1 CONTENTS       ="
        014322  012546                          MOV     (R5)+,-(SP)
        014324  012746  015226                  MOV     #XS1CON,-(SP)
        014330  012746  000002                  MOV     #2,-(SP)
        014334  010600                          MOV     SP,R0
        014336  104415                          TRAP    C$PNTX
        014340  062706  000006                  ADD     #6,SP
2693    014344                                  PRINTX  #XS2CON,(R5)+   ;PRINT "XSTAT2 CONTENTS       ="
        014344  012546                          MOV     (R5)+,-(SP)
        014346  012746  015273                  MOV     #XS2CON,-(SP)
        014352  012746  000002                  MOV     #2,-(SP)
        014356  010600                          MOV     SP,R0
        014360  104415                          TRAP    C$PNTX
        014362  062706  000006                  ADD     #6,SP
2694    014366                                  PRINTX  #XS3CON,(R5)+   ;PRINT "XSTAT3 CONTENTS       ="
        014366  012546                          MOV     (R5)+,-(SP)
        014370  012746  015340                  MOV     #XS3CON,-(SP)
        014374  012746  000002                  MOV     #2,-(SP)
        014400  010600                          MOV     SP,R0
        014402  104415                          TRAP    C$PNTX
        014404  062706  000006                  ADD     #6,SP
2695    014410  022737  000001  002134          CMP     #1,TRANSTST     ;CHECK FOR DUMP
2696    014416  001042                          BNE     50$             ;BR, IF NO DUMP REQUIRED
2697    014420                                  PRINTX  #RAMFHR
        014420  012746  014526                  MOV     #RAMFHR,-(SP)
        014424  012746  000001                  MOV     #1,-(SP)
        014430  010600                          MOV     SP,R0
        014432  104415                          TRAP    C$PNTX
        014434  062706  000004                  ADD     #4,SP
2698    014440  012737  000010  002246          MOV     #8.,RAMSIZ      ;RAM FIELD IS 8 BYTES LONG
```

```
2699 014446  012737  000020  011010          MOV     #20,RAMHLD      ;FIELD STARTS AT 20 OCTAL (10 HEX)
2700 014454  004737  010626                  JSR     PC,RAMER        ;READ AND PRINT THEM
2701 014460  012737  000040  011010          MOV     #40,RAMHLD      ;FIELD STARTS AT 40 OCTAL (20 HEX)
2702 014466  004737  010626                  JSR     PC,RAMER        ;READ AND PRINT THEM
2703 014472  012737  000060  011010          MOV     #60,RAMHLD      ;FIELD STARTS AT 60 OCTAL (30 HEX)
2704 014500  004737  010626                  JSR     PC,RAMER        ;READ AND PRINT THEM
2705 014504  012737  000020  002246          MOV     #16.,RAMSIZ     ;RAM FIELD IS SIXTEEN BYTES LONG
2706 014512  012737  000100  011010          MOV     #100,RAMHLD     ;FIELD STARTS AT 100 OCTAL (40 HEX)
2707 014520  004737  010626                  JSR     PC,RAMER        ;READ AND PRINT THEM
2708 014524  000207                  50$:    RTS     PC              ;RETURN
2709 014526    045    116    045  RAMFHR: .ASCIZ  '%N%A ********** SPECIAL M7454 RAM MEMORY DUMP **********'
2710 014617    045    116    045  MESBFN: .ASCIZ  '%N%A MESSAGE BUFFER CONTENTS PROBABLY NOT VALID'
2711 014677    045    116    045  PROASC: .ASCIZ  '%N%A Message Buffer Address = %01%05'
2712 014744    045    116    045  PR1ASC: .ASCIZ  '%N%A Message Buffer Contents:'
2713
2714 015002    045    116    045  MESHEA: .ASCIZ  '%N%A Message Buffer Header     = %06'
2715 015047    045    116    045  DATAFL: .ASCIZ  '%N%A Data Field Length         = %06'
2716 015114    045    116    045  RBPCRA: .ASCIZ  '%N%A Residual Byte Counter     = %06'
2717 015161    045    116    045  XS0CON: .ASCIZ  '%N%A XSTAT0 Contents           = %06'
2718 015226    045    116    045  XS1CON: .ASCIZ  '%N%A XSTAT1 Contents           = %06'
2719 015273    045    116    045  XS2CON: .ASCIZ  '%N%A XSTAT2 Contents           = %06'
2720 015340    045    116    045  XS3CON: .ASCIZ  '%N%A XSTAT3 Contents           = %06'
2721                                  .EVEN
```

```
                                        .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                                ;+[B
                                ;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
                                ;-
015406                          PRMSGEXP::
015406                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
015412  010005                          MOV     R0,R5           ;SAVE NUMBER OF WORDS
015414  013700  002252                  MOV     RCVLOADD,R0     ;GET RECV LOW ADDRESS
015420  010004                          MOV     R0,R4           ;COPY LOW ADDRESS
015422  013701  002250                  MOV     RCVHIADD,R1     ;GET RECV HIGH ADDRESS
015426  006100                          ROL     R0              ;SHIFT BIT15 TO C BIT
015430  006101                          ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
015432                                  PRINTX  #PRMSG0,R1,R4   ;PRINT MESSAGE BUFFER ADDRESS
015432  010446                          MOV     R4,-(SP)
015434  010146                          MOV     R1,-(SP)
015436  012746  015566                  MOV     #PRMSG0,-(SP)
015442  012746  000003                  MOV     #3,-(SP)
015446  010600                          MOV     SP,R0
015450  104415                          TRAP    C$PNTX
015452  062706  000010                  ADD     #10,SP
015456                                  PRINTX  #PRMSG1         ;PRINT HEADER FOR CONTENTS
015456  012746  015633                  MOV     #PRMSG1,-(SP)
015462  012746  000001                  MOV     #1,-(SP)
015466  010600                          MOV     SP,R0
015470  104415                          TRAP    C$PNTX
015472  062706  000004                  ADD     #4,SP
015476  005004                          CLR     R4              ;NUMBER OF THE CURRENT WORD
015500  012701  002266                  MOV     #EXPMSG,R1      ;GET EXPD BUFFER ADDRESS
015504  012702  002432                  MOV     #RECMSG,R2      ;GET RECV BUFFER ADDRESS
015510  011100                20$:      MOV     (R1),R0         ;GET EXPD
015512  011203                          MOV     (R2),R3         ;GET RECV
015514                                  XOR     R0,R3           ;XOR EXPD/RECV
015524                                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
015524  010346                          MOV     R3,-(SP)
015526  012246                          MOV     (R2)+,-(SP)
015530  012146                          MOV     (R1)+,-(SP)
015532  010446                          MOV     R4,-(SP)
015534  012746  015671                  MOV     #PRMSG2,-(SP)
015540  012746  000005                  MOV     #5,-(SP)
015544  010600                          MOV     SP,R0
015546  104415                          TRAP    C$PNTX
015550  062706  000014                  ADD     #14,SP
015554  005204                          INC     R4              ;NUMBER OF THE NEXT
015556  020405                          CMP     R4,R5           ;DONE ALL YET?
015560  002001                          BGE     50$             ;BR IF YES
015562  000752                          BR      20$             ;DO ANOTHER
015564  000207                50$:      RTS     PC              ;RETURN
015566    045   116   045     PRMSG0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
015633    045   116   045     PRMSG1: .ASCIZ  '%N%A Message Buffer Contents:'
015671    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
                ;-
015756          PRBYTEXP::
015756                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
015762  010005          MOV     R0,R5                   ;SAVE NUMBER OF BYTES
015764  005037  002264  CLR     PRMNO                   ;INIT ERROR COUNT
015770  005004          CLR     R4                      ;NUMBER OF THE CURRENT BYTE
015772  012701  002266  MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
015776  012702  002432  MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
016002  111100          20$:    MOVB    (R1),R0         ;GET EXPD BYTE
016004  042700  177400  BIC     #^C<377>,R0             ;CLEAR UPPER BYTE
016010  110037  016324  MOVB    R0,PRBEXP               ;SAVE FOR ERROR REPORT
016014  111203          MOVB    (R2),R3                 ;GET RECV BYTE
016016  042703  177400  BIC     #^C<377>,R3             ;CLEAR UPPER BYTE
016022  110337  016326  MOVB    R3,PRBREC               ;FOR ERROR REPORT
016026          XOR     R0,R3                           ;XOR EXPD/RECV
016036  122122          CMPB    (R1)+,(R2)+             ;EXPD = RECV?
016040  001431          BEQ     30$                     ;BR IF YES
016042  005237  002264  INC     PRMNO                   ;UPDATE ERROR COUNT
016046  023727  002264  000010  CMP     PRMNO,#8.       ;PRINTED 8?
016054  101023          BHI     30$                     ;BR IF YES
016056          27$:    PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
016056  010346          MOV     R3,-(SP)
016060  013746  016326  MOV     PRBREC,-(SP)
016064  013746  016324  MOV     PRBEXP,-(SP)
016070  010446          MOV     R4,-(SP)
016072  012746  016172  MOV     #PRBMSG,-(SP)
016076  012746  000005  MOV     #5,-(SP)
016102  010600          MOV     SP,R0
016104  104415          TRAP    C$PNTX
016106  062706  000014  ADD     #14,SP
016112          FORCEXIT        50$             ;aad
016122  000404          BR      35$             ;aad
016124          30$:
016124          FORCERROR  27$,NOTSSR           ;aad
016134          35$:                            ;aad
016134  005204          INC     R4              ;NUMBER OF THE NEXT
016136  020405          CMP     R4,R5           ;DONE ALL YET?
016140  002001          BGE     50$             ;BR IF YES
016142  000717          BR      20$             ;DO ANOTHER
016144          50$:    PRINTX  #PRBTOT,PRMNO   ;PRINT TOTAL ERROR COUNT
016144  013746  002264  MOV     PRMNO,-(SP)
016150  012746  016257  MOV     #PRBTOT,-(SP)
016154  012746  000002  MOV     #2,-(SP)
016160  010600          MOV     SP,R0
016162  104415          TRAP    C$PNTX
```

```
      016164  062706  000006                  ADD     #6,SP
 2803 016170  000207                          RTS     PC              ;RETURN
 2804
 2805 016172     045     116     045  PRBMSG: .ASCIZ  '%N%A   BYTE #%D2%A  EXPD: %03%A  RECV: %03%A  XOR: %03'
 2806 016257     045     116     045  PRBTOT: .ASCIZ  '%N%A NUMBER OF BYTES IN ERROR = %D2'
 2807                                  .EVEN
 2808 016324  000000                  PRBEXP: .WORD   0               ;EXPD
 2809 016326  000000                  PRBREC: .WORD   0               ;RECV
 2810
 2811                                  ;+
 2812                                  ;
 2813                                  ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
 2814                                  ;
 2815                                  ;INPUTS:
 2816                                  ;
 2817                                  ;       R1      RECEIVED DATA
 2818                                  ;       R2      EXPECTED DATA
 2819                                  ;
 2820                                  ;-
 2821 016330                          EXPREC: BGNMSG  EXPREC
      016330                          EXPREC::
 2822 016330  004737  007462                  JSR     PC,PRIXOR       ;PRINT THE DATA
 2823 016334                                  ENDMSG
      016334                          L10017:
      016334  104423                          TRAP    C$MSG
```

```
                                        .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
                                ;
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA BYTE
                                ;       R2      EXPECTED DATA BYTE
                                ;
                                ;-

016336                                  BGNMSG  EXPBREC
016336                          EXPBREC::
016336  004737  007332                  JSR     PC,PRIBXOR      ;PRINT THE DATA
016342                                  ENDMSG
016342                          L10020:
016342  104423                          TRAP    C$MSG


                                        .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
                                ;-

016344                                  BGNMSG  RAMERR
016344                          RAMERR::
016344  004737  013620                  JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
016350                                  ENDMSG
016350                          L10021:
016350  104423                          TRAP    C$MSG


                                        .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
016352                          RAMTADD::
016352  004737  010014                  JSR     PC,PRITADD      ;PRINT TEST ADDRESS
016356  004737  013620                  JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
016362                                  ENDMSG
016362                          L10022:
016362  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
016364                          RAMEXP::
016364  042701  177400                  BIC     #^C<377>,R1     ;SAVE EXPD RAM DATA BYTE
016370  042702  177400                  BIC     #^C<377>,R2     ;SAVE EXPD RAM DATA BYTE
016374  004737  007606                  JSR     PC,PRIRAM       ;PRINT THE RAM ADDRESS
016400  004737  007462                  JSR     PC,PRIXOR       ;PRINT THE DATA
016404                                  ENDMSG
016404                          L10023:
016404  104423                          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
```

```
 2927                                   ;-
 2928
 2929 016406                                    BGNMSG  TIMEXP
      016406                            TIMEXP::
 2930 016406                                    PRINTX  #TIMSG0         ;PRINT HEADER
      016406  012746  016434                    MOV     #TIMSG0,-(SP)
      016412  012746  000001                    MOV     #1,-(SP)
      016416  010600                            MOV     SP,R0
      016420  104415                            TRAP    C$PNTX
      016422  062706  000004                    ADD     #4,SP
 2931 016426  004737  007462                    JSR     PC,PRIXOR       ;PRINT THE DATA
 2932 016432                                    ENDMSG
      016432                            L10024:
      016432  104423                            TRAP    C$MSG
 2933
 2934
 2935 016434     045     116     045  TIMSG0: .ASCIZ  '%N%A TIMER A STATUS IS IN BIT 3%N%A TIMER B STATUS IS IN BIT 2'
 2936                                   .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)
                                ;
                                ;-

                                        BGNMSG  BADSSR
016534                          BADSSR::
016534  010246                          MOV     R2,-(SP)        ;SAVE DATA TRANSFERRED
016536  042702  177400                  BIC     #177400,R2      ;GET JUST ONE BYTE
016542                                  PRINTB  #XFERASC,R2
016542  010246                          MOV     R2,-(SP)
016544  012746  016574                  MOV     #XFERASC,-(SP)
016550  012746  000002                  MOV     #2,-(SP)
016554  010600                          MOV     SP,R0
016556  104414                          TRAP    C$PNTB
016560  062706  000006                  ADD     #6,SP
016564  012602                          MOV     (SP)+,R2        ;RESTORE R2
016566  004737  005264                  JSR     PC,PRITSSR      ;DECODE TSSR CONTENTS
016572                                  ENDMSG
016572                          L10025:
016572  104423                          TRAP    C$MSG
016574    045     116     0'S   XFERASC:        .ASCIZ  '%N%A Data Transferred = %03'
```

```
                                        .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
                                ;       JSR     PC,SOFINIT
                                ;       BCS     CONTINUE
                                ;       ERRDF                   ;REPORT FATAL ERROR
                                ;
                                ;-

016630                          SOFINIT::
016630                                  SAVREG                  ; SAVE THE REGISTERS
016634  012765  000000  000000          MOV     #0,TSSR(R5)     ; DO THE INIT.
016642  004737  017104                  JSR     PC,WAITF        ; WAIT FOR SSR
016646  016500  000000                  MOV     TSSR(R5),R0     ;GET THE TSSR REGISTER
016652  010004                          MOV     R0,R4           ;TSSR CONTENTS
016654  042704  176277                  BIC     #^C<HIADDR!OFL>,R4
016660  052704  002200                  BIS     #SSR!NBA,R4     ;R4 HAS EXPECTED CONTENTS
016664  020400                          CMP     R4,R0           ;ONLY EXPECTED BITS SET ?
016666  001402                          BEQ     5$              ;BRANCH IF OKAY
016670  000241                          CLC                     ;CLEAR THE CARRY FOR ERROR
016672  000401                          BR      10$             ;GO TO EXIT
016674  000261                  5$:     SEC                     ;SET THE CARRY BIT
016676  000207                  10$:    RTS     PC              ;RETURN TO CALLER
```

```
                                .SBTTL  CHKAMB  - CHECK TSSR FOR AMBIGUITY

                        ;+
                        ;
                        ;THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
                        ;FOR AMBIGUITY
                        ;
                        ;INPUT:
                        ;
                        ;       R0      CONTENTS OF TSSR
                        ;
                        ;OUTPUT:
                        ;
                        ;       R0      CONTENTS OF TSSR
                        ;
                        ;       CARRY   SET - NO AMBIGUITY
                        ;               CLR - AMBIGUOUS CONTENTS
                        ;
                        ;-

016700                  CHKAMB:
016700                          SAVREG                  ;SAVE THE GENERAL REGISTERS
016704  010004                  MOV     R0,R4           ;CONTENTS OF TSSR
016706  032700  100000          BIT     #SC,R0          ;IS BIT 15 SET ?
016712  001004                  BNE     5$              ;BRANCH IF YES
016714  032700  174077          BIT     #^C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
016720  001023                  BNE     40$             ;MUST BE AN ERROR
016722  000424                  BR      45$             ;RETURN WITH SUCCESS
016724  032700  000200  5$:     BIT     #SSR,R0         ;IS READY BIT SET ?
016730  001011                  BNE     10$             ;BRANCH IF READY BIT IS SET.
016732  032700  000040          BIT     #BIT5,R0        ;IS FATAL ERROR BIT SET ?
016736  001414                  BEQ     40$             ;ERROR IF NOT
016740  042704  177761          BIC     #^CTERCLS,R4    ;CLEAR ALL BUT TERMINATION CODE
016744  020427  000016          CMP     R4,#16          ;ALL THREE BITS MUST BE SET
016750  001007                  BNE     40$             ;ERROR IF NOT SET
016752  000410                  BR      45$             ;OK IF ALL ARE SET
016754  032700  000040  10$:    BIT     #BIT5,R0        ;IS FATAL ERROR BIT SET ?
016760  001405                  BEQ     45$             ;ERROR IF BIT IS SET WITH SSR
016762  032700  000006          BIT     #BIT2!BIT1,R0   ;IS THIS A FUNCTION REJECT
016766  001002                  BNE     45$             ;BR, IF TSSR IS OK
016770  000241          40$:    CLC                     ;AMBIGUOUS CONTENTS
016772  000401                  BR      50$
016774  000261          45$:    SEC                     ;SHOW SUCCESS - NO AMBIGUITY
016776  000207          50$:    RTS     PC              ;RETURN TO CALLER
```

```
                                        .SBTTL  ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
                                ;
                                ; DEFAULT DISPLAY INTERRUPT HANDLERS.
                                ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
                                ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
                                ;

                                ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
                                ;
  000200                            IOKCKIN=BIT7            ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
  000001                            IOKSTP=BIT0             ; EXPECT "STOP" INTERRUPT.
                                ;
                                ;INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
017000    000                   INTMASK:        .BYTE   0
                                ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
017001    000                   INTFLAG:        .BYTE   0

                                ;SAVED INTERRUPT VECTOR:
017002  000000                  INTVEC: .WORD   0
                                ;SAVE CPU PC
017004  000000                  INTCPC: .WORD   0

                                ;SUBROUTINE TO ENABLE INTERRUPTS:
017006  010046                  ENAINT: MOV     R0,-(SP)        ;SAVE R0
017010  013700  002156                  MOV     IVEC,R0         ;GET POINTER TO VECTORS
017014  012720  017052                  MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
017020  012720  000340                  MOV     #PRI07,(R0)+
017024  012600                          MOV     (SP)+,R0        ;RESTORE R0
017026  011646                          MOV     (SP),-(SP)
017030  012766  000000  000002          MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
017036  000002                          RTI

                                ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
017040  011646                  DSBINT: MOV     (SP),-(SP)
017042  012766  000340  000002          MOV     #PRI07,2(SP)
017050  000002                          RTI
```

```
                                                .SBTTL  INTR    - INTERRUPT HANDLERS

                                                BGNSRV  INTR            ;DEFINE INTERRUPT ENTRY
                                        INTR::
                                                MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
                                                CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
                                                BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
                                                BNE     1$              ;BR IF YES
                                                BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.

                                        ;SAVE REGISTERS, MSG BUFFER, ETC.
                                        1$:
                                                ENDSRV
                                        L10026:
                                                RTI
```

```
                                        .SBTTL  WAITF   - WAIT FOR SUBSYSTEM READY
                                ;
                                ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                ;
                                ;OUTPUTS:
                                ;
                                ;       R0      CONTENTS OF LAST TSSR READ
                                ;       CARRY   SET - READY BIT SET
                                ;               CLR - TIMEOUT WAITING FOR READY
                                ;
017104                          WAITF:: BREAK                           ; DO A SUPVSR BREAK FIRST.
017104  104422                          TRAP    C$BRK
017106  012746  010000                  MOV     #10000,-(SP)    ;BIG MSEC TIMER
017112                                  DELAY   1               ;DELAY 100US
017112  012727  000001                  MOV     #1,(PC)+
017116  000000                          .WORD   0
017120  013727  002116                  MOV     L$DLY,(PC)+
017124  000000                          .WORD   0
017126  005367  177772                  DEC     -6(PC)
017132  001375                          BNE     .-4
017134  005367  177756                  DEC     -22(PC)
017140  001367                          BNE     .-20
017142  016500  000000          2$:     MOV     TSSR(R5),R0     ;READ THE TSSR REGISTER
017146  105700                          TSTB    R0              ;TEST FOR READY BIT SET

017150  100420                          BMI     3$              ; EXIT ON STOP FLAG.
017152                                  DELAY   1               ; WAIT 100 USEC
017152  012727  000001                  MOV     #1,(PC)+
017156  000000                          .WORD   0
017160  013727  002116                  MOV     L$DLY,(PC)+
017164  000000                          .WORD   0
017166  005367  177772                  DEC     -6(PC)
017172  001375                          BNE     .-4
017174  005367  177756                  DEC     -22(PC)
017200  001367                          BNE     .-20
017202  005316                          DEC     (SP)            ;REDUCE DELAY COUNT
017204  001356                          BNE     2$              ;RETRY UNTIL TIMER EXPIRES
017206  000241                          CLC                     ; C = 0, CONTROLLER STILL RUNNING...
017210  000401                          BR      4$              ;...OR HUNG-UP AFTER 300 MSEC.
017212  000261                  3$:     SEC                     ; C = 1, CONTROLLER IS STOPPED.
017214  005326                  4$:     DEC     (SP)+           ;RESTORE STACK WITHOUT CHANGING CARRY BIT
017216  000207                          RTS     PC
```

```
                        .SBTTL  CHKTSSR - CHECK TSSR FOR READY

                ;+
                ;
                ;THIS ROUTINE WAITS FOR READY IN THE TSSR
                ;AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
                ;
                ;INPUT:
                ;
                ;       R5      ADDRESS OF CSR REGISTERS
                ;
                ;OUTPUT:
                ;
                ;       R0      CONTENTS OF TSSR
                ;       CARRY   SET - OKAY
                ;               CLR - NOT READY AMBIGUOUS, OR SC SET
                ;
                ;-

017220          CHKTSSR:
017220  004737  017104          JSR     PC,WAITF        ;WAIT FOR READY
017224  103014                  BCC     20$             ;BRANCH IF TIME OUT
017226  004737  016700          JSR     PC,CHKAMB       ;TSSR AMBIGOUS?
017232  103006                  BCC     10$             ;BR IF YES
017234  032700  100000          BIT     #SC,R0          ;SPECIAL CONDITION SET?
017240  001405                  BEQ     15$             ;BR IF NO
017242  032700  074000          BIT     #<SCE!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
017246  001402                  BEQ     15$             ;BR IF NO
017250  000241          10$:    CLC                     ;SET FAILURE
017252  000401                  BR      20$             ;
017254  000261          15$:    SEC                     ;SET SUCCESS
017256  000207          20$:    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.
                                ;
017260  012737  017312  000004  XNXM:   MOV     #2$,@#4         ; SET BUSERR VECTOR.
017266  012737  000200  000006          MOV     #PRI04,@#6
017274  005003                          CLR     R3              ;FLAG.
017276  005711                  1$:     TST     (R1)            ;TEST THE ADDRESS(ES).
                                                                ;IF ANY TRAP, CONTINUE AT 2$.
017300  020102                          CMP     R1,R2           ;OTHERWISE, CONTINUE HERE.
017302  001407                          BEQ     3$              ;BR IF FINISHED (NO NEXM'S).
017304  062701  000002                  ADD     #2,R1           ;SET NEXT ADDRESS...
017310  000772                          BR      1$              ;...AND CONTINUE.

017312  005103                  2$:     COM     R3              ;GOT ONE, SET FLAG...
017314  012716  017322                  MOV     #3$,(SP)
017320  000002                          RTI                     ;...AND DISMISS INTERRUPT...
017322                          3$:     CLRVEC  #4              ;...AND GIVE BACK THE VECTOR.
017322  012700  000004                  MOV     #4,R0
017326  104436                          TRAP    C$CVEC
017330  005703                          TST     R3              ;DID WE CATCH ONE ??
017332  001401                          BEQ     .+4             ;NO, "C" = 0, SKIP NEXT.
017334  000261                          SEC                     ;YES, "C" = 1, (R1) = NEXM ADDR.
017336  000207                          RTS     PC



                                        .SBTTL  TSTLOOP - CHECK ITERATION COUNT
                                ;+
                                ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
                                ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
                                ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
                                ;
                                ; CALL: LOOPTO  ARG
                                ;
017340                          TSTLOOP::
017340  005737  002136                  TST     NOITS           ; ITERATIONS INHIBITED?
017344  001006                          BNE     1$              ; YES.
017346  005737  002152                  TST     QVP             ; NO.
017352  100403                          BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
017354  005337  002164                  DEC     LOOPCNT         ; BUMP LOOP COUNTER.
017360  001002                          BNE     2$
017362  000241                  1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
017364  000401                          BR      3$
017366  000261                  2$:     SEC                     ;LOOP ENABLED.
017370  000207                  3$:     RTS     PC
```

```
                                        .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
                                ;+
                                ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
                                ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
                                ; IN THE CURRENT RUN SEQUENCE.
                                ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      POINTER TO TEST ID ASCIZ STRING
                                ;
                                ;OUTPUT:
                                ;
                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       INTERRUPT LEVEL IS RASIED TO LEVEL OF
                                ;       THE DEVICE UNDER TEST
                                ;
                                ;-
                                ;
                                TSTSETUP::
017372  010046                          MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
017374  005037  003106                   CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
017400  005037  017640                   CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
017404  005037  005232                   CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
017410  105037  017000                   CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
017414  013700  002150                   MOV     UNITN,R0        ; GET THE UNIT NUMBER,
017420  006300                           ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
017422  005737  003062                   TST     NODEV           ; DID STARTUP FIND THE DEVICE?
017426  001430                           BEQ     4$              ; BR IF YES
017430  100010                           BPL     3$              ; BR IF NOT IDLE
017432  052760  160000  003130           BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
017440                                   ERRDF   1,NXR,NXRERR    ; NO DEVICE HERE -- PRINT IT
017440  104455                           TRAP    C$ERDF
017442  000001                           .WORD   1
017444  003636                           .WORD   NXR
017446  005176                           .WORD   NXRERR
017450  000407                           BR      2$
017452  052760  160001  003130  3$:      BIS     #160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
017460                                   ERRDF   2,NOINIT        ; DEVICE NOT IDLE
017460  104455                           TRAP    C$ERDF
017462  000002                           .WORD   2
017464  004233                           .WORD   NOINIT
017466  000000                           .WORD   0
017470  012737  177777  003060  2$:      MOV     #-1,DUFLG       ; DROP THE UNIT
017476                                   DODU    UNITN
017476  013700  002150                   MOV     UNITN,R0
017502  104451                           TRAP    C$DODU
017504                                   DOCLN                   ; ABORT THE PASS
```

```
                017504  104444                          TRAP    C$DCLN
3265            017506  000423                          BR      5$
3266
3267            017510                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
                017510  104421                          TRAP    C$RFLA
3268            017512  032700  001000                  BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
3269            017516  001412                          BEQ     1$              ; BR IF NO
3270            017520  011600                          MOV     (SP),R0         ;GET THE ID MESSAGE
3271            017522                                  PRINTF  #TNAM,R0        ;DISPLAY THE TEST ID
                017522  010046                          MOV     R0,-(SP)
                017524  012746  017566                  MOV     #TNAM,-(SP)
                017530  012746  000002                  MOV     #2,-(SP)
                017534  010600                          MOV     SP,R0
                017536  104417                          TRAP    C$PNTF
                017540  062706  000006                  ADD     #6,SP
3272            017544  005237  002162          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
3273            017550                                  SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
                017550  013700  002160                  MOV     IPRI,R0
                017554  104441                          TRAP    C$SPRI
3274            017556  005726                  5$:     TST     (SP)+           ;FIX UP THE STACK
3275            017560  013705  002154                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
3276            017564  000207                          RTS     PC
3277            017566     045     123     045  TNAM:  .ASCIZ  '%S%T%A Test'
3278                                                    .EVEN
```

```
                                        .SBTTL  TSTEND  - PRINT ERRORS RECEIVED
                                ;
                                ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
                                ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
                                ;
017602                          TSTEND: RFLAGS  R0
017602  104421                         TRAP    C$RFLA
017604  030027  020000                 BIT     R0,#IER
017610  001412                         BEQ     1$                      ; BR IF "IER" NOT SET.
017612                                 PRINTF  #ESUM,ERRK              ; PRINT ERROR COUNT.
017612  013746  017640                 MOV     ERRK,-(SP)
017616  012746  017642                 MOV     #ESUM,-(SP)
017622  012746  000002                 MOV     #2,-(SP)
017626  010600                         MOV     SP,R0
017630  104417                         TRAP    C$PNTF
017632  062706  000006                 ADD     #6,SP
017636  000207                 1$:     RTS     PC

017640  000000                 ERRK:   0                               ; LOCAL ERROR COUNT.
017642     045     101     040 ESUM:   .ASCIZ  /%A %D%A ERRORS/
017661     105     122     122 EMAXDU: .ASCIZ  /ERROR LIMIT REACHED -- DROPPING UNIT/
                                        .EVEN

                                        .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
                                ;+
                                ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
                                ;-
017726  005237  017640         INCERK: INC     ERRK                    ; INCREMENT LOCAL ERROR COUNT
017732  010046                         MOV     R0,-(SP)                ; SAVE R0
017734  013700  002150                 MOV     UNITN,R0                ; GET UNIT NUMBER,
017740  006300                         ASL     R0                      ; ... AND MAKE IT A WORD OFFSET.
017742  062700  003130                 ADD     #ERTABL,R0              ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
017746  005210                         INC     (R0)                    ; INCREMENT THE DEVICE ERROR COUNT
017750  032710  007777                 BIT     #7777,(R0)              ; DID WE OVERFLOW THE FIELD?
017754  001001                         BNE     1$                      ; BR IF NO.
017756  005310                         DEC     (R0)                    ; YES -- BACK IT UP TO 7777.
017760  012600                 1$:     MOV     (SP)+,R0                ; RESTORE R0
017762  000207                         RTS     PC                      ; RETURN TO CALLER.

017764  010046                 CKEMAX: MOV     R0,-(SP)                ; SAVE R0
017766  013700  002150                 MOV     UNITN,R0                ; GET UNIT NUMBER
017772  006300                         ASL     R0                      ; ... AND MAKE IT A WORD OFFSET
017774  016000  003130                 MOV     ERTABL(R0),R0           ; GET ERROR TABLE ENTRY
020000  042700  170000                 BIC     #170000,R0              ; EXTRACT ERROR COUNT FIELD
020004  020037  002142                 CMP     R0,GERRMAX              ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
020010  103004                         BHIS    1$                      ; BR IF YES
020012  023737  017640  002140         CMP     ERRK,LERRMAX            ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
020020  103417                         BLO     2$                      ; BR IF NO
020022                           1$:     RFLAGS  R0                      ; GET OPERATOR FLAGS
020022  104421                         TRAP    C$RFLA
020024  032700  000040                 BIT     #IDU,R0                 ; IS DPOPPING INHIBITED?
020030  001013                         BNE     2$                      ; BR IF YES.
020032  012737  177777  003060         MOV     #-1,DUFLG               ; NO -- DROP THE UNIT
020040                                 ERRDF   4,EMAXDU
020040  104455                         TRAP    C$ERDF
020042  000004                         .WORD   4
020044  017661                         .WORD   EMAXDU
```

```
                020046  000000                          .WORD   0
     3326       020050                                  DODU    UNITN
                020050  013700  002150                  MOV     UNITN,R0
                020054  104451                          TRAP    C$DODU
     3327       020056                                  DOCLN
                020056  104444                          TRAP    C$DCLN
     3328       020060  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
     3329       020062  000207                          RTS     PC              ; RETURN TO CALLER
     3330                                               .SBTTL  FATCHK - INC FATAL ERRORS AND CHECK FOR LIMIT
     3331                                           ;+
     3332                                           ;
     3333                                           ;       CHECK FATAL COUNTER, AFTER INC, FOR MORE THAN 25
     3334                                           ;       ERRORS AND IF OVER CALL UNIT DROP ROUTINE
     3335                                           ;
     3336                                           ;-
     3337       020064                              FATCHK:
     3338       020064                                      SAVREG                          ;BETTER SAVE THE REGISTERS
     3339       020070  013701  002150                      MOV     UNITN,R1                ;PICK UP THE UNIT NUMBER
     3340       020074  006301                              ASL     R1                      ;MAKE IT INTO A BYTE OFFSET
     3341       020076  062761  000001  003130              ADD     #1,ERTABL(R1)           ;ADD 1 TO THE PROPER UNIT'S ERROR COUNTER
     3342       020104  005237  002170                      INC     FATFLG                  ;BUMP FATAL ERROR COUNTER
     3343       020110  023727  002170  000031              CMP     FATFLG,#25.             ;CHECK AGAINST 25
     3344       020116  002406                              BLT     9$                      ;BR, IF LESS THAN 25 ERRORS
     3345       020120                                      RFLAGS  R0                      ;READ THE FLAGS INTO R0
                020120  104421                              TRAP    C$RFLA
     3346       020122  032700  040000                      BIT     #BIT14,R0               ;BR, IF LOOP ON ERROR IS SET
     3347       020126  001002                              BNE     9$                      ;OTHERWISE NEVER BE ABLE TO SCOPE ETC.
     3348       020130  004737  020136                      JSR     PC,CKDROP               ;DROP UNIT IF ALLOWED
     3349       020134  000207                  9$:     RTS     PC                      ;RETURN ETC.
     3350                                           ;
     3351                                           ;
     3352                                           ;
```

```
                                        .SBTTL  CKDROP  - CHECK IF UNIT SHOULD BE DROPPED
                                ;+
                                ; CHECK IF UNIT SHOULD BE DROPPED
                                ;-
020136  010046                 CKDROP: MOV     R0,-(SP)
020140                                  FORCERROR       1$,NOTSSR
020150                                  RFLAGS  R0
020150  104421                         TRAP    C$RFLA
020152  032700  000040                 BIT     #IDU,R0
020156  001010                         BNE     1$
020160  011600                         MOV     (SP),R0
020162  012737  177777  003060         MOV     #-1,DUFLG
020170                                  DODU    UNITN
020170  013700  002150                 MOV     UNITN,R0
020174  104451                         TRAP    C$DODU
020176                                  DOCLN                   ;ABORT THE PASS
020176  104444                         TRAP    C$DCLN
020200  012600                 1$:     MOV     (SP)+,R0
020202  000207                         RTS     PC


                                        .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
                                ;
                                ; SUBROUTINE - DETERMINE CONFIGURATION OF TU80 SYSTEM.
                                ;
020204                          CONFIG:
020204  004737  016630                 JSR     PC,SOFINIT
020210  000207                         RTS     PC
```

```
3384                                        .SBTTL  KTON,KTOFF      - ENABLE/DISABLE MEMORY MANAGEMENT
3385                                ;
3386                                ; SUBROUTINE - ENABLE MEM MGT.
3387                                ;
3388 020212  005737  003100        KTON:   TST     KTFLG           ; GOT KT?
3389 020216  001403                        BEQ     1$              ; NO.
3390 020220  012737  000001  177572        MOV     #1,SR0          ; YES. ENABLE KT11.
3391 020226  000207                1$:     RTS     PC
3392
3393
3394
3395                                ;
3396                                ; SUBROUTINE - DISABLE MEM MGT.
3397                                ;
3398 020230  005737  003100        KTOFF:  TST     KTFLG           ; GOT KT11?
3399 020234  001405                        BEQ     1$              ; NO.
3400 020236  000240                        NOP
3401 020240  000240                        NOP
3402 020242  012737  000000  177572        MOV     #0,SR0          ; DISABLE KT.
3403 020250  000207                1$:     RTS     PC
3404
3405
```

```
                                .SBTTL  SETMAP  - SETUP PAR6 MAPPING

                        ;+
                        ;
                        ;THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
                        ;AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
                        ;IS RETURNED BIASED TO PAR6.
                        ;
                        ;INPUTS:
                        ;
                        ;       R0      HIGH ORDER ADDRESS BITS
                        ;       R1      LOW ORDER ADDRESS BITS
                        ;
                        ;OUTPUTS:
                        ;
                        ;       R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
                        ;       CARRY   SET IF SUCCESS
                        ;               CLR IF ERROR
                        ;-
020252                  SETMAP:
020252                          SAVREG                  ;SAVE R1-R4 UNTIL NEXT RETURN
020256  005737  003100          TST     KTFLG           ;SYSTEM HAVE ABOVE 28K?
020262  001433                  BEQ     10$             ;BR IF NO
020264  010102                  MOV     R1,R2           ;SAVE LOW ORDER BITS
        000006                  .REPT   6
                                ASR     R0              ;CONVERT WORD ADDRESS TO 32W BLOCKS
                                ROR     R1              ;MAKE IT DUUBLE PRECISION
                                .ENDR
020316  042701  000177          BIC     #177,R1         ;ALINE FOR LOWER 4K BOUNDARY
020322  020137  003100          CMP     R1,KTFLG        ;HIGHER THAN EXISTING MEMORY?
020326  103011                  BHIS    10$             ;BR IF YES
020330  010137  172354          MOV     R1,@#KIPAR6     ;SETUP MAPPING REGISTER PAR6
020334  042702  160000          BIC     #160000,R2      ;SETUP DISPLACEMENT IN PAGE
020340  062702  140000          ADD     #140000,R2      ;ADD IN PAR6 BIAS
020344  010200                  MOV     R2,R0           ;RETURN IN R0
020346  000261                  SEC                     ;SET SUCCESS
020350  000401                  BR      15$             ;
020352  000241          10$:    CLC                     ;SET FAILURE
020354  000207          15$:    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
                                ;-
                                ;
020356                          FILLMEM:
020356                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
020362  004737  020230                   JSR     PC,KTOFF        ;DISABLE KT.
020366  010003                           MOV     R0,R3           ;COPY TEST PATTERN
020370  013701  003072                   MOV     FREE,R1         ;GET FIRST FREE LOCATION
020374  013702  003074                   MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
020400  010321                  10$:     MOV     R3,(R1)+        ;STORE A BACKGROUND WORD
020402  005302                           DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
020404  003375                           BGT     10$             ;BR IF NO
020406  005737  003100                   TST     KTFLG           ; GOT KT?
020412  001452                           BEQ     55$             ; NO. GET OUT.
020414  004737  020212                   JSR     PC,KTON         ; YES. ENABLE KT.
020420  005000                           CLR     R0              ;HIGH ORDER ADDRESS START
020422  013701  003104                   MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                           .REPT   6
                                         CLC                     ;CLEAR C BIT
                                         ROL     R1              ;CONVERT BLOCKS TO WORDS
                                         ROL     R0              ;MAKE IT DOUBLE PRECISION
                                         .ENDR
020472  004737  020252                   JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020476  010320                  30$:     MOV     R3,(R0)+        ;STORE TEST PATTERN IN >28K ADDRESS
020500  020027  160000                   CMP     R0,#160000      ;END OF PAR6 MAPPING AREA?
020504  103774                           BLO     30$             ;BR IF NO
020506  162700  020000                   SUB     #20000,R0       ;BACKUP INTO PAR6 MAPPING BEGIN
020512  062737  000200  172354           ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
020520  023737  172354  003100           CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
020526  001402                           BEQ     50$             ;BR IF YES
020530  000137  020476                   JMP     30$             ;KEEP GOING ON ETC.
020534  004737  020230          50$:     JSR     PC,KTOFF        ; DISABLE KT.
020540  000207                  55$:     RTS     PC
```

```
                                        .SBTTL  CMPMEM  - COMPARE MEMORY TO BACKGROUND PATTERN
                                ;+
                                ; COMPARE MEMORY WITH A BACKGROUND PATTERN
                                ;
                                ; INPUTS:
                                ;
                                ;       R0 = BACKGROUND PATTERN
                                ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                                ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       CARRY   - SET IF NO ERROR
                                ;       CARRY   - CLR IF ERROR
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       ERRHI   - ERROR HIGH ADDRESS
                                ;       ERRLO   - ERROR LOW ADDRESS
                                ;       EXPD    - EXPECTED DATA
                                ;       RECV    - RECEIVED DATA
                                ;-
020542                          CMPMEM:
020542                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
020546  010003                          MOV     R0,R3           ;COPY TEST PATTERN
020550  004737  020230                  JSR     PC,KTOFF        ;DISABLE KT.
020554  013701  003072                  MOV     FREE,R1         ;GET FIRST FREE LOCATION
020560  013702  003074                  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
020564  020311                  10$:    CMP     R3,(R1)         ;FREE SPACE LOCATION EQUAL TO EXPD?
020566  001411                          BEQ     15$             ;BR IF YES
020570  010137  002204                  MOV     R1,ERRLO        ;SAVE ADDRESS IN ERROR
020574  005037  002202                  CLR     ERRHI           ;NO HIGH ADDRESS
020600  010337  002176                  MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
020604  011137  002200                  MOV     (R1),RECV       ;SAVE RECV FOR ERROR REPORT
020610  000474                          BR      50$             ;
020612  005721                  15$:    TST     (R1)+           ;POINT TO NEXT ADDRESS
020614  005302                          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
020616  003362                          BGT     10$             ;BR IF NO
020620  005737  003100                  TST     KTFLG           ; GOT KT?
020624  001472                          BEQ     55$             ; NO. GET OUT.
020626  004737  020212                  JSR     PC,KTON         ; YES. ENABLE KT.
020632  005000                          CLR     R0              ;HIGH ORDER ADDRESS START
020634  013701  003104                  MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                          .REPT   6
                                        ROL     R1              ;CONVERT BLOCKS TO WORDS
                                        ROL     R0              ;MAKE IT DOUBLE PRECISION
                                        .ENDR
020670  042701  000177                  BIC     #177,R1         ;ALINE 4K BOUNDARY
020674  010046                          MOV     R0,-(SP)        ;SAVE HIGH ORDER
020676  010146                          MOV     R1,-(SP)        ;SAVE LOW ORDER
020700  004737  020252                  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020704  010004                          MOV     R0,R4           ;COPY ADDRESS BIASED TO PAR6
020706  012601                          MOV     (SP)+,R1        ;RESTORE LOW ORDER IN NON PAR6 FORMAT
020710  012600                          MOV     (SP)+,R0        ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
020712  020314                  30$:    CMP     R3,(R4)         ;ABOVE 28K LOCATION EQUAL EXPD?
020714  001411                          BEQ     32$             ;BR IF YES
020716  010037  002202                  MOV     R0,ERRHI        ;SAVE HIGH ORDER IN ERROR
```

```
3553 020722  010137  002204                 MOV     R1,ERRLO        ;SAVE LOW ORDER IN ERROR
3554 020726  010337  002176                 MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
3555 020732  011437  002200                 MOV     (R4),RECV       ;SAVE RECV FOR ERROR REPORT
3556 020736  000421                         BR      50$             ;
3557 020740  062701  000002          32$:   ADD     #2,R1           ;UPDATE NON PAR6 ADDRESS
3558 020744  005500                         ADC     R0              ;MAKE IT DOUBLE PRECISION ADD
3559 020746  062704  000002                 ADD     #2,R4           ;UPDATE PAR FORMAT ADDRESS
3560 020752  020427  160000                 CMP     R4,#160000      ;END OF PAR6 MAPPING AREA?
3561 020756  103755                         BLO     30$             ;BR IF NO
3562 020760  162704  020000                 SUB     #20000,R4       ;BACKUP INTO PAR6 MAPPING BEGIN
3563 020764  062737  000200  172354         ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
3564 020772  023737  172354  003100         CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
3565 021000  101744                         BLOS    30$             ;BR IF NO
3566 021002  004737  020230          50$:   JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
3567 021006  000241                         CLC                     ;SET FAILURE
3568 021010  000403                         BR      60$             ;
3569 021012  004737  020230          55$:   JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
3570 021016  000261                         SEC                     ;SET SUCCESS
3571 021020  000207                 60$:    RTS     PC
3572
```

```
                                        .SBTTL  REGSAV  - SAVE R1-R5 ON STACK
                                ;+
                                ;
                                ;ROUTINE TO
                                ;SAVE R1 THROUGH R5 ON THE STACK
                                ;
                                ;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
                                ;
                                ;-

021022                          REGSAV:
021022                                  BREAK                   ;LOOK FOR CNTL C
021022  104422                          TRAP    C$BRK
021024  010446                          MOV     R4,-(SP)
021026  010346                          MOV     R3,-(SP)
021030  010246                          MOV     R2,-(SP)
021032  010146                          MOV     R1,-(SP)
021034  010546                          MOV     R5,-(SP)
021036  016605  000012                  MOV     10.(SP),R5
021042  004736                          JSR     PC,@(SP)+
021044  012601                          MOV     (SP)+,R1
021046  012602                          MOV     (SP)+,R2
021050  012603                          MOV     (SP)+,R3
021052  012604                          MOV     (SP)+,R4
021054  012605                          MOV     (SP)+,R5
021056                                  BREAK                   ;LOOK FOR CNTL C
021056  104422                          TRAP    C$BRK
021060  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
                                ;
                                ;-
                                GETPAT::
021062                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
021062                          1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
021066  104443                          TRAP    C$GMAN
021070  000406                          BR      10000$
021072  021116                          .WORD   PATDAT
021074  000022                          .WORD   T$CODE
021076  021120                          .WORD   DATASC
021100  000377                          .WORD   377
021102  000000                          .WORD   T$LOLIM
021104  000377                          .WORD   T$HILIM
021106                          10000$:
021106                                  BNCOMPLETE      1$      ;RETRY IF ERROR
021106  103367                          BCC     1$
021110  013700  021116                  MOV     PATDAT,R0       ;DATA PATTERN FROM OPERATOR
021114  000207                          RTS     PC              ;RETURN TO CALLER

                                ;+
                                ;LOCAL DATA AREA
                                ;-

021116  000000                 PATDAT: .WORD   0               ;TEMPORARY STORAGE FOR DATA
021120     105     116     124 DATASC: .ASCIZ  'ENTER DATA PATTERN'
                                    .EVEN
```

```
                                        .SBTTL  GETSEL  - ISSUE MENU AND GET OPERATOR RESPONSE
                                ;+
                                ;ROUTINE TO ISSUE A MENU AND GET
                                ;THE OPERATOR'S RESPONSE.
                                ;INPUTS:
                                ;       R0      ADDRESS OF ASCIZ STRING OF MENU
                                ;       R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
                                ;OUTPUTS:
                                ;       R0      NUMBER OF THE OPERATOR'S SELECTION
                                ;-
021144                          GETSEL::
021144                                  SAVREG                          ;SAVE GENERAL REGISTERS
021150  010002                          MOV     R0,R2                   ;SAVE THE MENU ADDRESS
021152  010203                  1$:     MOV     R2,R3                   ;START OF MENU STRING
021154  005713                  2$:     TST     (R3)                    ;END OF ASCII ?
021156  001412                          BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
021160                                  PRINTF  #SELASC,(R3)+           ;DISPLAY THE MENU
021160  012346                          MOV     (R3)+,-(SP)
021162  012746  021330                  MOV     #SELASC,-(SP)
021166  012746  000002                  MOV     #2,-(SP)
021172  010600                          MOV     SP,R0
021174  104417                          TRAP    C$PNTF
021176  062706  000006                  ADD     #6,SP
021202  000764                          BR      2$
021204                          3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
021204  104443                          TRAP    C$GMAN
021206  000406                          BR      10001$
021210  021364                          .WORD   MENRES
021212  000042                          .WORD   T$CODE
021214  021335                          .WORD   MENASC
021216  177777                          .WORD   -1
021220  000000                          .WORD   T$LOLIM
021222  177777                          .WORD   T$HILIM
021224                          10001$:
021224                                  BNCOMPLETE      1$      ;RETRY IF ERROR
021224  103352                          BCC     1$
021226  013700  021364                  MOV     MENRES,R0       ;GET THE OPERATOR'S REPLY
021232  020001                          CMP     R0,R1           ;COMPARE TO MAXIMUM ALLOWED
021234  101411                          BLOS    5$              ;BRANCH IF OK
021236                                  PRINTF  #MENERR         ;DISPLAY ERROR MESSAGE
021236  012746  021262                  MOV     #MENERR,-(SP)
021242  012746  000001                  MOV     #1,-(SP)
021246  010600                          MOV     SP,R0
021250  104417                          TRAP    C$PNTF
021252  062706  000004                  ADD     #4,SP
021256  000735                          BR      1$              ;RETRY
021260  000207                  5$:     RTS     PC              ;RETURN TO CALLER
021262     045     116     045  MENERR: .ASCIZ  '%N%A *** Menu Selection Too Large ***'
021330     045     116     045  SELASC: .ASCIZ  '%N%T'
021335     105     156     164  MENASC: .ASCIZ  'Enter Menu Selection: '
                                        .EVEN
021364  000000                  MENRES:         .WORD   0
```

```
                                        .SBTTL  CHKMAN  - CHECK MANUAL INTERVENTION LEGALITY
                                ;+
                                ;
                                ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
                                ;
                                ;INPUT:
                                ;
                                ;       NONE.
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   0       MANUAL INTERVENTION NOT ALLOWED
                                ;               1       MANUAL INTERVENTION IS OK
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       A MESSAGE IS DISPLAYED WARNING THAT TEST IS
                                ;       NOT EXECUTED IF MANUAL INTERVENTION IS NOT
                                ;       ALLOWED.
                                ;
                                ;-

021366                          CHKMAN::
021366                                  SAVREG                  ;SAVE THE REGISTERS
021372                                  MANUAL                  ;SEE IF MANUAL INTERVENTION OK
021372  104450                          TRAP    CSMANI
021374                                  BCOMPLETE 1$            ;BRANCH IF ALLOWED
021374  103411                          BCS     1$
021376                                  PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
021376  012746  021422                  MOV     #NOMAN,-(SP)
021402  012746  000001                  MOV     #1,-(SP)
021406  010600                          MOV     SP,R0
021410  104417                          TRAP    CSPNTF
021412  062706  000004                  ADD     #4,SP
021416  000241                          CLC                     ;CLEAR CARRY FOR ERROR
021420  000207                  1$:     RTS     PC              ;RETURN

021422      045     116     045 NOMAN:  .ASCIZ  '%N%A *** Manual Intervention not Allowed - Test Aborted ***'
                                        .even
```

```
3711                                            .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
3712                                    ;
3713                                    ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
3714                                    ;
3715 021516                            ENVIRN: MEMORY  R0
     021516  104431                             TRAP    C$MEM
3716 021520  010037  003072                    MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
3717 021524  062737  000002  003072            ADD     #2,FREE
3718 021532  011037  003074                    MOV     (R0),FRESIZ     ;...AND WORD COUNT.
3719 021536  162737  000004  003074            SUB     #4,FRESIZ
3720 021544  013702  002012                    MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
3721 021550  162737  000007  003074  10$:      SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
3722 021556  005302                            DEC     R2
3723 021560  001373                            BNE     10$
3724 021562  013700  003072                    MOV     FREE,R0         ;GET FIRST FREE ADDRESS
3725 021566  063700  003074                    ADD     FRESIZ,R0       ;POINT TO LAST FREE ADDRESS
3726 021572  162700  000002                    SUB     #2,R0           ;BACKUP 1 WORD
3727 021576  010037  003076                    MOV     R0,FREEHI       ;STORE LAST FREE ADDRESS
3728 021602  000207                            RTS     PC              ;RETURN
3729
```

```
                                        .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
                                ;+
                                ;
                                ;ROUTINE TO INIT KT-11
                                ;
                                ;-

021604                          KTINIT:
021604  005037  003100                  CLR     KTFLG           ; INIT >28K MEMORY FLAG
021610  005037  003102                  CLR     KTENABLE        ; INIT TEST >28K FLAG
021614  023727  002120  001577          CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
021622  101444                          BLOS    9$              ; NO.
021624  013700  000004                  MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
021630  012737  021722  000004          MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
021636  005737  177572                  TST     @#SR0           ; GOT KT11?
021642  000240                          NOP                     ;  (TRAP IF NO).
021644  013737  002120  003100          MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
021652  042737  000177  003100          BIC     #177,KTFLG      ;
021660  010037  000004                  MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
021664  005000                          CLR     R0              ; R0 = AR DATA.
021666  012701  172340                  MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
021672  012761  077406  177740  1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
021700  010021                          MOV     R0,(R1)+        ; SET KIPAR REG.
021702  062700  000200                  ADD     #200,R0         ; BUMP AR DATA BY "4K".
021706  020027  002000                  CMP     R0,#2000        ; AT "I/O"?
021712  001367                          BNE     1$              ; NO.
021714  012741  177600                  MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
021720  000405                          BR      9$
                                ;
021722  012716  021730          2$:     MOV     #6$,(SP)        ; SET UP RETURN
021726  000002                          RTI                     ; RTI TO NEXT LOCATION
                                ;
021730  010037  000004          6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
                                ;
021734  000207                  9$:     RTS     PC
                                ;
                                ;
                                        BGNPROT
021736                          LSPROT::
021736  177777  177777  177777          .WORD   -1, -1, -1, -1  ;NO DEVICE PROTECTION REQUIRED.
021746                                  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
                                ;-
                                        BGNINIT
021746                          L$INIT::
                                40$:
021746  012737  005755  002146          MOV     #EPRT1,EPRTSW   ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021754  005037  003106                  CLR     SIFLAG          ;CLEAR "SOFT INIT" FLAG
021760  005037  003102                  CLR     KTENABLE        ;CLEAR TEST ABOVE 28K FLAG
021764  005037  002246                  CLR     RAMSIZ          ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021770                                  READEF  #EF.CONTINUE
021770  012700  000036                  MOV     #EF.CONTINUE,R0
021774  104447                          TRAP    C$REFG
021776                                  BNCOMPLETE 1$
021776  103023                          BCC     1$
022000  023737  002150  002012          CMP     UNITN,L$UNIT    ;UNIT IN RANGE?
022006  103073                          BHIS    4$              ;BR IF NO.
022010  005737  003060                  TST     DUFLG           ;DROPPED UNIT?
022014  100475                          BMI     NXTU            ;BR IF YES
022016  013701  002150                  MOV     UNITN,R1
022022  006301                          ASL     R1
022024  005761  003130                  TST     ERTABL(R1)
022030  001521                          BEQ     SETU
022032  032761  040000  003130          BIT     #BIT14,ERTABL(R1) ;DROPPED?
022040  001063                          BNE     NXTU
022042                                  EXIT    INIT            ;DO NOTHING IF "CONTINUE".
022042  104432                          TRAP    C$EXIT
022044  000430                          .WORD   L10030-.
022046                          1$:     READEF  #EF.NEW
022046  012700  000035                  MOV     #EF.NEW,R0
022052  104447                          TRAP    C$REFG
022054                                  BNCOMPLETE NXTU         ;TAKE NEXT UNIT IF NOT NEW PASS.
022054  103055                          BCC     NXTU
022056                                  READEF  #EF.START
022056  012700  000040                  MOV     #EF.START,R0
022062  104447                          TRAP    C$REFG
022064                                  BCOMPLETE 2$
022064  103404                          BCS     2$
022066                                  READEF  #EF.RESTART
022066  012700  000037                  MOV     #EF.RESTART,R0
022072  104447                          TRAP    C$REFG
022074                                  BNCOMPLETE 31$
022074  103034                          BCC     31$
022076                          2$:                             ;1ST PASS, BUS-INIT...
022076                                  BRESET                  ;BUS RESET.
022076  104433                          TRAP    C$RESET
022100  005037  002162                  CLR     TSTCNT          ;NUMBER OF TESTS RUN IN PASS
022104  005037  002170                  CLR     FATFLG          ;RESET FLAG TO ZERO "FATAL ERRORS"
```

```
3827 022110  000406                          BR      19$             ;BR, IF THE FLAG IS NOT SET
3828                                                                 ;(NO DEBUGGER ETC.)
3829 022112  012746  000340                  MOV     #340,-(SP)
3830 022116  012746  022132                  MOV     #20$,-(SP)      ;RETURN TO DEBUGGER
3831 022122  000137  062046                  JMP     O.ODT           ;;@@ENTER THE DEBUGGER
3832 022126  005037  003332          19$:    CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
3833 022132                          20$:
3834 022132  012737  177777  002152          MOV     #-1,QVP         ;...QUICK VERIFY...
3835 022140  004737  021516                  JSR     PC,ENVIRN       ;SET ENVIRONMENT.
3836 022144  004737  021604                  JSR     PC,KTINIT       ;INITIALIZE KT MEMORY MANAGEMENT
3837 022150  012700  003130                  MOV     #ERTABL,R0
3838 022154  005020                  30$:    CLR     (R0)+           ;CLEAR THE ERROR TABLE
3839 022156  020027  003330                  CMP     R0,#ERTABE
3840 022162  103774                          BLO     30$
3841 022164  000404                          BR      4$
3842 022166  005037  002152          31$:    CLR     QVP
3843 022172  000137  022242                  JMP     PASRPT          ;GO REPORT THE STATUS
3844
3845 022176                          4$:
3846 022176  012737  177777  002150  NEWPAS: MOV     #-1,UNITN       ;INIT UNIT NUMBER...
3847 022204  005037  002166                  CLR     DEVCNT          ;CLEAR COUNT OF DEVICES RUNNING
3848 022210                          NXTU:   BREAK
     022210  104422                          TRAP    C$BRK
3849 022212  005237  002150                  INC     UNITN           ;...AND SET NEXT UNIT NUMBER.
3850 022216  023737  002150  002012          CMP     UNITN,L$UNIT
3851 022224  103423                          BLO     SETU
3852 022226  012737  177777  003060          MOV     #-1,DUFLG
3853 022234  000401                          BR      11$
3854 022236                                  DOCLN                   ;ABORT, NO MORE UNITS.
     022236  104444                          TRAP    C$DCLN
3855 022240  000240                  11$:    NOP
3856 022242                          PASRPT:
3857 022242  023727  002012  000001          CMP     L$UNIT,#1       ;HOW MANY UNITS SELECTED?
3858 022250  101752                          BLOS    NEWPAS          ;BR IF ONLY 1
3859 022252  005737  002166                  TST     DEVCNT          ;ARE ANY STILL RUNNING?
3860 022256  001747                          BEQ     NEWPAS          ;BR IF NO
3861 022260                                  RFLAGS  R0
     022260  104421                          TRAP    C$RFLA
3862 022262  032700  000100                  BIT     #ISR,R0         ;SHOULD WE PRINT STATISTICS
3863 022266  001343                          BNE     NEWPAS          ;BR IF NO
3864
3865 022270                                  DORPT
     022270  104424                          TRAP    C$DRPT
3866 022272  000741                          BR      NEWPAS
3867 022274                          10$:
3868
3869 022274                          SETU:   GPHARD  UNITN,R0        ;GET UNIT N P-TABLE POINTER.
     022274  013700  002150                  MOV     UNITN,R0
     022300  104442                          TRAP    C$GPHRD
3870 022302                                  BNCOMPLETE NXTU         ;BR IF UNIT NOT AVAILABLE.
     022302  103342                          BCC     NXTU
3871 022304  005037  003060                  CLR     DUFLG           ;CLEAR "DROPPED" FLAG.
3872 022310  005237  002166                  INC     DEVCNT
3873 022314  012001                          MOV     (R0)+,R1        ;GET 1ST REGISTER ADDRESS.
3874 022316  010137  002154                  MOV     R1,CSRADDR      ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
3875
3876 022322  012001                          MOV     (R0)+,R1        ;GET VECTOR ADDRESS.
```

```
                                        MOV     (R0),R2         ;GET INTERRUPT PRIORITY
                                        MOV     R2,IPRI         ;SET INTERRUPT PRIORITY.
                                        MOV     R1,IVEC         ;SET INTERRUPT VECTOR POINTER...
                                        MOV     #INTR,(R1)+     ;...VECTOR...
                                        MOV     R2,(R1)+        ;...AND PRIORITY.

                                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.
                                ;
                                        MOV     UNITN,R1
                                        ASL     R1
                                        BIS     #BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
                                        CLR     EXTA            ;CLEAR ERROR EXTENSION FLAG.
                                        CMP     L$UNIT,#1       ;ARE WE TESTING MULTIPLE UNITS?
                                        BLOS    10$             ;BR IF NO.
                                        RFLAGS  R0              ;YES -- GET OPERATOR FLAGS.
                                        TRAP    C$RFLA
                                        BIT     #PNT,R0         ;SHOULD WE PRINT UNIT #?
                                        BEQ     10$             ;BR IF NOT.
                                        PRINTF  #PUNIT,UNITN    ;PRINT THE UNIT #
                                        MOV     UNITN,-(SP)
                                        MOV     #PUNIT,-(SP)
                                        MOV     #2,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTF
                                        ADD     #6,SP
                                10$:
                                        CLR     NODEV
                                        MOV     CSRADDR,R1      ;ADDRESS OF FIRST REGISTER
                                        MOV     R1,R2           ;START OF REGISTERS
                                        ADD     #TSSR,R2        ;ADDRESS OF TSSR REGISTER
                                        JSR     PC,XNXM         ;TEST BOTH CONTROLLER REGISTERS...
                                        BCC     2$              ;...AND BR IF ALL OK.
                                        MOV     R1,NODEV        ;FLAG DEVICE AS NON-EXISTENT
                                        MOV     #-1,DUFLG       ;DROP THIS UNIT.
                                2$:
                                ;
                                ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
                                ;
                                5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
                                        MOV     #PRI00,R0
                                        TRAP    C$SPRI
                                        ENDINIT
                                L10030:
                                        TRAP    C$INIT

                                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.
                                ;--
022544                          BGNAU
022544                  L$AU::
022544 010001                   MOV     R0,R1           ; GET UNIT TO BE ADDED (R0)
022546 006301                   ASL     R1              ; MAKE IT A WORD INDEX
022550 052761 100000 003130     BIS     #100000,ERTABL(R1) ; SET THE "ACTIVE" BIT
022556 042761 040000 003130     BIC     #40000,ERTABL(R1)  ; CLEAR THE "DROPPED" BIT
022564                          PRINTF  #1$,R0
022564 010046                   MOV     R0,-(SP)
022566 012746 022612            MOV     #1$,-(SP)
022572 012746 000002            MOV     #2,-(SP)
022576 010600                   MOV     SP,R0
022600 104417                   TRAP    C$PNTF
022602 062706 000006            ADD     #6,SP
022606                          EXIT    AU
022606 000167                   .WORD   J$JMP
022610 000026                   .WORD   L10031-2-.
022612    045    116    045  1$:      .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN

022640                          ENDAU                   ; UNUSED.
022640                  L10031:
022640 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.

022642                          BGNDU
022642                  L$DU::
022642 012737 177777 003060     MOV     #-1,DUFLG
022650 010001                   MOV     R0,R1
022652 006301                   ASL     R1
022654 052761 140000 003130     BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022662 000240 000240 000240     240,240,240                     ; ????????
022670                          PRINTF  #1$,R0
022670 010046                   MOV     R0,-(SP)
022672 012746 022716            MOV     #1$,-(SP)
022676 012746 000002            MOV     #2,-(SP)
022702 010600                   MOV     SP,R0
022704 104417                   TRAP    C$PNTF
022706 062706 000006            ADD     #6,SP
022712                          EXIT    DU
022712 000167                   .WORD   J$JMP
022714 000030                   .WORD   L10032-2-.
```

```
022716      045      116      045  1$:      .ASCIZ  /%N%A  UNIT %D%A DROPPED/
                                            .EVEN
022746                                      ENDDU
022746                             L10032:
022746  104453                              TRAP    C$DU
                                   ;++
                                   ; AUTO-DROP CODE SECTION.
                                   ;--
022750                                      BGNAUTO
022750                             L$AUTO::
022750  012703  000550                      MOV     #360.,R3        ;ENOUGH TIME FOR 2400' REEL TO REWIND
022754  004737  017104             10$:     JSR     PC,WAITF        ;WAIT FOR SSR TO SET
022760  103420                              BCS     20$             ;LEAVE WHEN SSR IS SET
022762                                      DELAY   250.            ;WAIT FOR .25 SECONDS
022762  012727  000372                      MOV     #250.,(PC)+
022766  000000                              .WORD   0
022770  013727  002116                      MOV     L$DLY,(PC)+
022774  000000                              .WORD   0
022776  005367  177772                      DEC     -6(PC)
023002  001375                              BNE     .-4
023004  005367  177756                      DEC     -22(PC)
023010  001367                              BNE     .-20
023012  005303                              DEC     R3              ;BUMP COUNTER DOWN
023014  001357                              BNE     10$             ;KEEP GOING
023016  004737  020136                      JSR     PC,CKDROP       ;TRY AND DROP UNIT
023022                             20$:
023022                                      ENDAUTO                 ; UNUSED.
023022                             L10033:
023022  104461                              TRAP    C$AUTO
```

```
                                        .SBTTL  CLEAN-UP AND REPORT CODING SECTIONS

                                ;++
                                ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
                                ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
                                ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
                                ;--
023024                                  BGNCLN
023024                          L$CLEAN::
023024  005737  003060                  TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
023030  100405                          BMI     1$              ;...AND GROSS CONTROLLER FAULT...
                                                                ;...DON'T TRY TO XCT CLEANUP CODE.

023032  012765  000000  000000          MOV     #0,TSSR(R5)     ;DO SOFT INIT
023040  004737  017104                  JSR     PC,WAITF
023044                          1$:
023044                          2$:     ENDCLN
023044                          L10034:
023044  104412                          TRAP    C$CLEAN
                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--
023046                                  BGNRPT
023046                          L$RPT::
023046                                  PRINTS  #DEVSUM
023046  012746  023310                  MOV     #DEVSUM,-(SP)
023052  012746  000001                  MOV     #1,-(SP)
023056  010600                          MOV     SP,R0
023060  104416                          TRAP    C$PNTS
023062  062706  000004                  ADD     #4,SP
023066  010246                          MOV     R2,-(SP)
023070  010346                          MOV     R3,-(SP)
023072  010446                          MOV     R4,-(SP)
023074  012704  003130                  MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
023100  005003                          CLR     R3              ; CLEAR UNIT NUMBER
023102  011402                  1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
023104  001467                          BEQ     4$              ; ZERO IF UNIT NOT RUN
023106  100066                          BPL     4$
023110  032702  040000                  BIT     #BIT14,R2       ; WAS UNIT DROPPED?
023114  001015                          BNE     2$              ; BR IF YES
023116  042702  170000                  BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
023122                                  PRINTS  #DEVONL,R3,R2   ; PRINT
023122  010246                          MOV     R2,-(SP)
023124  010346                          MOV     R3,-(SP)
023126  012746  023345                  MOV     #DEVONL,-(SP)
023132  012746  000003                  MOV     #3,-(SP)
023136  010600                          MOV     SP,R0
023140  104416                          TRAP    C$PNTS
023142  062706  000010                  ADD     #10,SP
023146  000446                          BR      4$
023150  020227  160000          2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
023154  001012                          BNE     3$              ; BR IF NO
023156                                  PRINTS  #DEVNXR,R3
023156  010346                          MOV     R3,-(SP)
023160  012746  023415                  MOV     #DEVNXR,-(SP)
023164  012746  000002                  MOV     #2,-(SP)
```

```
                023170  010600                          MOV     SP,R0
                023172  104416                          TRAP    C$PNTS
                023174  062706  000006                  ADD     #6,SP
        4012    023200  000431                          BR      4$
        4013    023202  020227  160001          3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
        4014    023206  001012                          BNE     30$             ; BR IF NO.
        4015    023210                                  PRINTS  #DEVNRD,R3
                023210  010346                          MOV     R3,-(SP)
                023212  012746  023477                  MOV     #DEVNRD,-(SP)
                023216  012746  000002                  MOV     #2,-(SP)
                023222  010600                          MOV     SP,R0
                023224  104416                          TRAP    C$PNTS
                023226  062706  000006                  ADD     #6,SP
        4016    023232  000414                          BR      4$
        4017    023234  042702  170000          30$:    BIC     #^C7777,R2
        4018    023240                                  PRINTS  #DEVDRO,R3,R2
                023240  010246                          MOV     R2,-(SP)
                023242  010346                          MOV     R3,-(SP)
                023244  012746  023560                  MOV     #DEVDRO,-(SP)
                023250  012746  000003                  MOV     #3,-(SP)
                023254  010600                          MOV     SP,R0
                023256  104416                          TRAP    C$PNTS
                023260  062706  000010                  ADD     #10,SP
        4019    023264  062704  000002          4$:     ADD     #2,R4
        4020    023270  005203                          INC     R3
        4021    023272  020427  003330                  CMP     R4,#ERTABE
        4022    023276  103701                          BLO     1$
        4023    023300  012604                          MOV     (SP)+,R4
        4024    023302  012603                          MOV     (SP)+,R3
        4025    023304  012602                          MOV     (SP)+,R2
        4026    023306                                  ENDRPT                  ; UNUSED.
                023306                          L10035:
                023306  104425                          TRAP    C$RPT
        4027
        4028
        4029    023310      045     116     045 DEVSUM: .ASCIZ   /%N%ADEVICE STATUS SUMMARY:%N/
        4030    023345      045     101     040 DEVONL: .ASCIZ   /%A  UNIT %D3%A   ONLINE,  ERRORS = %D%N/
        4031    023415      045     101     040 DEVNXR: .ASCIZ   /%A  UNIT %D3%A   DROPPED, NON-EXISTENT REGISTER%N/
        4032    023477      045     101     040 DEVNRD: .ASCIZ   /%A  UNIT %D3%A   DROPPED, NOT READY AT STARTUP%N/
        4033    023560      045     101     040 DEVDRO: .ASCIZ   /%A  UNIT %D3%A   DROPPED, ERRORS = %D%N/
        4034                                            .EVEN
```

```
                                .SBTTL  TEST  1: FIFO EXERCISER
                        ;++
                        ; TEST DESCRIPTION:
                        ;
                        ;       This test uses the Write Subsystem Memory command to
                        ;       verify the controller's FIFO and associated status and
                        ;       control logic.
                        ;
                        ; TEST STEPS:
                        ;
                        ;  REPEAT FOR LOOPCNT
                        ;    BEGIN
                        ;       Do Subtest 1    - FIFO Initialize status test
                        ;       Do Subtest 2    - FIFO Write Single Byte test
                        ;       Do Subtest 3    - FIFO Write Multiple Bytes test
                        ;       Do Subtest 4    - FIFO Verify ILW Status test
                        ;       Do Subtest 5    - FIFO Input Ready test
                        ;       Do Subtest 6    - FIFO Verify Reset FIFO test
                        ;    END
                        ;--


023630                          BGNTST                                          T1::
023630
005037  002170                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
005037  003100                  CLR     KTFLG                   ;HOLD OF KT11
012700  031216                  MOV     #TST17ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
004737  017372                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
012737  000002  002164          MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
004737  020230                  JSR     PC,KTOFF                ;SHUT OFF MEMORY MANAGEMENT
005037  003102                  CLR     KTENABLE                ;REALLY SHUT DOWN KT-11
                        ;
                        ;
                        ;       THIS CODE CHECKS FOR THE DRIVE TO BE AT BOT VIA A REWIND COMMAND
                        ;
                        ;
023666                          BGNSUB                                          T1.1:
023666
023666  104402                                                                  TRAP    C$BSUB
012737  014544  033126          MOV     #6500.,T17DLY           ;SET UP LOOP COUNTER
                        ;*****************************************************************************
                        ;
                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF THERE'S AN ERROR
                        ;
                        ;*****************************************************************************
004737  016630          5$:     JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
103424                          BCS     10$                     ;BR IF INIT WAS OK
023704                          DELAY   250                     ;DELAY A WHILE
012727  000250                                                          MOV     #250,(PC)+
000000                                                                  .WORD   0
013727  002116                                                          MOV     L$DLY,(PC)+
000000                                                                  .WORD   0
005367  177772                                                          DEC     -6(PC)
```

```
        023724  001375                                                          BNE     .-4
        023726  005367  177756                                                  DEC     -22(PC)
        023732  001367                                                          BNE     .-20
 4115   023734  004737  020064          JSR     PC,FATCHK               ;INCREMENT AND CHECK FOR MORE THAN 25 ERRORS
 4116   023740  016501  000000          MOV     TSSR(R5),R1             ;CONTENTS OF THE TSSR REGISTER
 4117   023744                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
        023744  104455                                                          TRAP    C$ERDF
        023746  000144                                                          .WORD   100
        023750  003550                                                          .WORD   SFIERR
        023752  011656                                                          .WORD   SFIMSG
 4118   023754                  10$:
 4119
 4120   023754  012704  032640          MOV     #T17PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
 4121
 4122                           ;*************************************************************************************
 4123                           ;
 4124                           ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
 4125                           ;
 4126                           ;*************************************************************************************
 4127
 4128   023760  004737  010322          JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 4129   023764  103407                  BCS     15$                     ;BR IF COMMAND ISSUED IS OK
 4130   023766  004737  020064          JSR     PC,FATCHK               ;INCREMENT AND CHECK FOR MORE THAN 25 ERRORS
 4131   023772  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4135   023774                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTICS FAILED
        023774  104456                                                          TRAP    C$ERHRD
        023776  000145                                                          .WORD   101
        024000  004754                                                          .WORD   WRTMSG
        024002  011656                                                          .WORD   SFIMSG
 4136
 4137
 4138
 4139                           ;*************************************************************************************
 4140                           ;
 4141                           ;ISSUE A REWIND COMMAND TO THE SELECTED TAPE DRIVE
 4142                           ;
 4143                           ;*************************************************************************************
 4144
 4145   024004  004737  010424  15$:    JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
 4146   024010  103412                  BCS     30$                     ;BR IF NO PROBLEM
 4147   024012  005337  033126          DEC     T17DLY                  ;DEC DELAY COUNTER
 4148   024016  001327                  BNE     5$                      ;BR IF LOOP IS REQUIRED
 4149   024020  010001                  MOV     R0,R1                   ;SAVE TSSR
 4150   024022  004737  020064          JSR     PC,FATCHK               ;INCREMENT AND CHECK FOR MORE THAN 25 ERRORS
 4154   024026                          ERRHRD  ERRNO,T17RWN,PKTSSR     ;REWIND NOT ACCEPTED
        024026  104456                                                          TRAP    C$ERHRD
        024030  000146                                                          .WORD   102
        024032  031235                                                          .WORD   T17RWN
        024034  011670                                                          .WORD   PKTSSR
 4155   024036                  30$:    CKLOOP                          ;LOOP IF SELECTED
        024036  104406                                                          TRAP    C$CLP1
 4156   024040  104406                  TRAP    C$CLP1
 4157
 4158   024042  000137  030762          JMP     TMPEND                  ;aaaTEMP JUMP OVER FIFO TESTS
 4159   024046                  T17LOOP:
```

```
                                        .SBTTL  TEST  1: SUBTEST 1: FIFO INITIALIZE STATUS TEST
                                ;++
                                ; TEST 1: SUBTEST 1:
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This test verifies, by using the Read Status select code,
                                ;       that the FIFO status is in the correct initial state after
                                ;       the controller is initialized (Input Ready TRUE,
                                ;       Output Ready and Data In Miss FALSE). These status
                                ;       signals are checked by the controller's self-test
                                ;       sequence, so this subtest is actually more of a partial
                                ;       check of the Read Status function than the FIFO status.
                                ;
                                ; TEST STEPS:
                                ;
                                ;    BEGIN
                                ;       Write to TSSR to soft initialize
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                                ;       Do a WRITE SUBSYSTEM Read Status
                                ;       If Input Ready  NOT=1   Then Print Error
                                ;       If Output Ready NOT=0   Then Print Error
                                ;       If Data In Miss NOT=0   Then Print Error
                                ;    END
                                ;--
                                ;
                                ;       Write to TSSR register to soft initialize the controller
024046                          5$:
024046  004737  016630                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
024052  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
024054  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024056                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
024056  104455                                                                  TRAP    C$ERDF
024060  000146                                                                  .WORD   102
024062  003550                                                                  .WORD   SFIERR
024064  011656                                                                  .WORD   SFIMSG
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
024066  005037  002170          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
024072  012704  032640                  MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
024076  004737  010322                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
024102  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
024104  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024106                                  NEXT.ERRNO
024106                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
024106  104455                                                                  TRAP    C$ERDF
024110  000147                                                                  .WORD   103
024112  031263                                                                  .WORD   T17SSR
024114  011670                                                                  .WORD   PKTSSR
024116  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
024122                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
024122  104406                                                                  TRAP    C$CLP1
                                ;       Do a Write Subsystem READ STATUS
024124  004737  032422                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
024130  012704  033010                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
024134  010465  177776                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
024140  004737  017220                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
```

```
4209 024144  103407                        BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
4210 024146  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4211 024150                                NEXT.ERRNO
4212 024150                          62$:  ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     024150  104455                                                                TRAP    C$ERDF
     024152  000150                                                                .WORD   104
     024154  031364                                                                .WORD   T173SSR
     024156  011670                                                                .WORD   PKTSSR
4213 024160  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4214 024164                          70$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     024164  104406                                                                TRAP    C$CLP1
4215                                 ;     Set WORDS 0-7  of expd message buffer = to recv since not testing
4216 024166  004737  032604                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
4217 024172  012701  031012                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
4218 024176  012702  032702                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
4219 024202  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
4220 024204  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
4221 024206  052711  000020                BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
4222 024212  042711  000040                BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
4223 024216  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
4224                                 ;     If Input Ready NOT=1 then Print Error
4225                                 ;     If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
4226 024222  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4227 024224  012701  032662                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4228 024230  012702  030772                MOV     #T17EXP,R2              ;EXPD ADDRESS
4229 024234  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4230 024240  004737  011322                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4231 024244  103404                        BCS     90$                     ;BR IF YES
4232 024246                                NEXT.ERRNO
4233 024246                          82$:  ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
     024246  104456                                                                TRAP    C$ERHRD
     024250  000151                                                                .WORD   105
     024252  031603                                                                .WORD   T171CMP
     024254  012172                                                                .WORD   MSGSTAT
4234 024256                          90$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     024256  104406                                                                TRAP    C$CLP1
4235
4236 024260                                ENDSUB                          ;////////////// END SUBTEST //////////
     024260                                                                L10037:
     024260  104403                                                                TRAP    C$ESUB
4237
4238 024262  005737  002170                TST     FATFLG                  ;ANY FATAL ERRORS ?
4239 024266  001402                        BEQ     160$                    ;BRANCH IF NOT
4240 024270  004737  020136                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4241 024274                          160$:
4242
```

```
                                .SBTTL  TEST  1: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

                        ;++
                        ; TEST 1: SUBTEST 2:
                        ;
                        ; SUBTEST DESCRIPTION:
                        ;
                        ;       This subtest verifies the ability of the FIFO to correctly
                        ;       pass a single data byte from input to output. For each
                        ;       of 256 data values (0-377 octal) the following is done:
                        ;               1. Initial FIFO status is checked
                        ;               2. The Write FIFO function, specifying a count of
                        ;                  one byte to be written is executed.
                        ;               3. Read Status is executed and FIFO status is checked.
                        ;               4. Read FIFO is executed and the data and final status
                        ;                  is checked.
                        ;
                        ; TEST STEPS:
                        ;
                        ; BEGIN
                        ;       Write to TSSR to soft initialize
                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=0   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;
                        ; REPEAT FOR DATA FROM 0 TO 377 OCTAL
                        ;   BEGIN
                        ;       Do a Write Subsystem WRITE NPR to set tape direction out
                        ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=1   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;       Do Write Subsystem READ FIFO with byte count equal to 1
                        ;       If Data read from FIFO NOT= to Data sent Then Print Error
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=0   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;   END
                        ; END
                        ;--
024274                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
024274                                                                  T1.2:
024274  104402                                                                  TRAP    C$BSUB

                        ;       Write to TSSR register to soft initialize the controller
                        5$:
024276  004737  016630          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
024302  103405          BCS     10$                             ;BR IF SOFT INIT OKAY
024304  010001          MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
024306                  ERRDF   ERRNO,SFIERR,SFIMSG             ;DEVICE FATAL DURING INIT
024306  104455                                                                  TRAP    C$ERDF
024310  000151                                                                  .WORD   105
024312  003550                                                                  .WORD   SFIERR
```

```
        024314  011656                                                          .WORD   SFIMSG
 4296                           ;       Do a WRITE CHARACTERISTICS to setup a message buffer
 4297 024316  005037  002170    10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 4298 024322  012704  032640            MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 4299 024326  004737  010322            JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 4300 024332  103407                    BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 4301 024334  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4302 024336                            NEXT.ERRNO
 4303 024336                    42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        024336  104455                                                          TRAP    C$ERDF
        024340  000152                                                          .WORD   106
        024342  031263                                                          .WORD   T17SSR
        024344  011670                                                          .WORD   PKTSSR
 4304 024346  004737  020064            JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 4305 024352                    50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        024352  104406                                                          TRAP    C$CLP1
 4306                           ;       Do a Write Subsystem READ STATUS
 4307 024354  004737  032422            JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
 4308 024360  012704  033010            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4309 024364  010465  177776            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4310 024370  004737  017220            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4311 024374  103407                    BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
 4312 024376  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4313 024400                            NEXT.ERRNO
 4314 024400                    62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        024400  104455                                                          TRAP    C$ERDF
        024402  000153                                                          .WORD   107
        024404  031364                                                          .WORD   T173SSR
        024406  011670                                                          .WORD   PKTSSR
 4315 024410  004737  020064            JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 4316 024414                    70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        024414  104406                                                          TRAP    C$CLP1
 4317                           ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
 4318 024416  004737  032604            JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
 4319 024422  012701  031012            MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 4320 024426  012702  032702            MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 4321 024432  012221                    MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 4322 024434  011211                    MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 4323 024436  052711  000020            BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
 4324 024442  042711  000040            BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
 4325 024446  042711  000200            BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
 4326                           ;       If Input Ready NOT=1 then Print Error
 4327                           ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
 4328 024452  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 4329 024454  012701  032662            MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 4330 024460  012702  030772            MOV     #T17EXP,R2              ;EXPD ADDRESS
 4331 024464  012703  000024            MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 4332 024470  004737  011322            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 4333 024474  103404                    BCS     90$                     ;BR IF YES
 4334 024476                            NEXT.ERRNO
 4335 024476                    82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
        024476  104456                                                          TRAP    C$ERHRD
        024500  000154                                                          .WORD   108
        024502  031603                                                          .WORD   T171CMP
        024504  012172                                                          .WORD   MSGSTAT
 4336 024506                    90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        024506  104406                                                          TRAP    C$CLP1
```

```
4337
4338                                    ;  Repeat for DATA from 0 to 377
4339 024510  012737  000000  002256            MOV     #0,DATA                 ;GET FIRST DATA
4340 024516                             100$:                                   ;REPEAT LABEL
4341                                    ;       Do a Write Subsystem WRITE NPR to set tape direction out
4342 024516  012700  000100                    MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
4343 024522  004737  032464                    JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
4344 024526  012704  033010                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4345 024532  010465  177776                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4346 024536  004737  017220                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4347 024542  103407                            BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
4348 024544  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4349 024546                                    NEXT.ERRNO
4350 024546                             102$:  ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     024546  104455                                                                    TRAP    C$ERDF
     024550  000155                                                                    .WORD   109
     024552  031431                                                                    .WORD   T174SSR
     024554  011670                                                                    .WORD   PKTSSR
4351 024556  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4352 024562                             105$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     024562  104406                                                                    TRAP    C$CLP1
4353                                    ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
4354 024564  012700  000001                    MOV     #1,R0                   ;WRITE 1 BYTE
4355 024570  012701  002256                    MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4356 024574  004737  032510                    JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
4357 024600  012704  033010                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4358 024604  010465  177776                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4359 024610  004737  017220                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4360 024614  103407                            BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
4361 024616  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4362 024620                                    NEXT.ERRNO
4363 024620                             107$:  ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     024620  104455                                                                    TRAP    C$ERDF
     024622  000156                                                                    .WORD   110
     024624  031474                                                                    .WORD   T175SSR
     024626  011670                                                                    .WORD   PKTSSR
4364 024630  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4365 024634                             110$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     024634  104406                                                                    TRAP    C$CLP1
4366
4367                                    ;       Do a Write Subsystem READ STATUS
4368 024636  004737  032422                    JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
4369 024642  012704  033010                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4370 024646  010465  177776                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4371 024652  004737  017220                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4372 024656  103407                            BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
4373 024660  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4374 024662                                    NEXT.ERRNO
4375 024662                             112$:  ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     024662  104455                                                                    TRAP    C$ERDF
     024664  000157                                                                    .WORD   111
     024666  031364                                                                    .WORD   T173SSR
     024670  011670                                                                    .WORD   PKTSSR
4376 024672  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4377 024676                             120$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     024676  104406                                                                    TRAP    C$CLP1
4378                                    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
```

```
 4379 024700  004737  032604                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
 4380 024704  012701  031012                    MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 4381 024710  012702  032702                    MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 4382 024714  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 4383 024716  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 4384 024720  052711  000020                    BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
 4385 024724  052711  000040                    BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
 4386 024730  042711  000200                    BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
 4387                               ;           If Input ReadyB NOT=1 then Print Error
 4388                               ;           If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
 4389 024734  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 4390 024736  012701  032662                    MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 4391 024742  012702  030772                    MOV     #T17EXP,R2              ;EXPD ADDRESS
 4392 024746  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 4393 024752  004737  011322                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 4394 024756  103404                            BCS     140$                    ;BR IF YES
 4395 024760                                    NEXT.ERRNO
 4396 024760                        132$:       ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
      024760  104456                                                                    TRAP    C$ERHRD
      024762  000160                                                                    .WORD   112
      024764  031761                                                                    .WORD   T173CMP
      024766  012172                                                                    .WORD   MSGSTAT
 4397 024770                        140$:       CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      024770  104406                                                                    TRAP    C$CLP1
 4398
 4399                               ;           Do Write Subsystem READ FIFO with byte count equal to 1
 4400 024772  012700  000001                    MOV     #1,R0                   ;SET READ BYTE COUNT
 4401 024776  004737  032544                    JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
 4402 025002  012704  033010                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4403 025006  010465  177776                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4404 025012  004737  017220                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4405 025016  103407                            BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
 4406 025020  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4407 025022                                    NEXT.ERRNO
 4408 025022                        142$:       ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      025022  104455                                                                    TRAP    C$ERDF
      025024  000161                                                                    .WORD   113
      025026  031540                                                                    .WORD   T176SSR
      025030  011670                                                                    .WORD   PKTSSR
 4409 025032  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 4410 025036                        150$:       CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      025036  104406                                                                    TRAP    C$CLP1
 4411                               ;           Set WORDS 0-7 of expd message buffer = to recv since not testing
 4412 025040  004737  032604                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
 4413 025044  012701  031012                    MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 4414 025050  012702  032702                    MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 4415 025054  013721  002256                    MOV     DATA,(R1)+              ;SET EXPD WORD #8 = COUNT DATA
 4416 025060  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV (NOT TESTING)
 4417                               ;           If Data read from FIFO NOT= to Data sent Then Print Error
 4418                               ;           The data is in WORD #8 of the message buffer
 4419 025062  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 4420 025064  012701  032662                    MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 4421 025070  012702  030772                    MOV     #T17EXP,R2              ;EXPD ADDRESS
 4422 025074  012703  000022                    MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
 4423 025100  004737  011322                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 4424 025104  103404                            BCS     160$                    ;BR IF YES
 4425 025106                                    NEXT.ERRNO
```

```
4426 025106                          152$:   ERRHRD  ERRNO,T172CMP,MSGSUB    ;REPORT ERROR
     025106  104456                                                                TRAP    C$ERHRD
     025110  000162                                                                .WORD   114
     025112  031665                                                                .WORD   T172CMP
     025114  013564                                                                .WORD   MSGSUB
4427 025116                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     025116  104406                                                                TRAP    C$CLP1
4429                                 ;       Do a Write Subsystem READ STATUS
4430 025120  004737  032422                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
4431 025124  012704  033010                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4432 025130  010465  177776                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4433 025134  004737  017220                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4434 025140  103407                          BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
4435 025142  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4436 025144                                  NEXT.ERRNO
4437 025144                          162$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     025144  104455                                                                TRAP    C$ERDF
     025146  000163                                                                .WORD   115
     025150  031364                                                                .WORD   T173SSR
     025152  011670                                                                .WORD   PKTSSR
4438 025154  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4439 025160                          170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     025160  104406                                                                TRAP    C$CLP1
4440                                 ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
4441 025162  004737  032604                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
4442 025166  012701  031012                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
4443 025172  012702  032702                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
4444 025176  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
4445 025200  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
4446 025202  052711  000020                  BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
4447 025206  042711  000040                  BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
4448 025212  042711  000200                  BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
4449                                 ;       If Input Ready NOT=1 then Print Error
4450                                 ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
4451 025216  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4452 025220  012701  032662                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4453 025224  012702  030772                  MOV     #T17EXP,R2              ;EXPD ADDRESS
4454 025230  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4455 025234  004737  011322                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4456 025240  103404                          BCS     180$                    ;BR IF YES
4457 025242                                  NEXT.ERRNO
4458 025242                          172$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     025242  104456                                                                TRAP    C$ERHRD
     025244  000164                                                                .WORD   116
     025246  032045                                                                .WORD   T174CMP
     025250  012172                                                                .WORD   MSGSTAT
4459 025252                          180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     025252  104406                                                                TRAP    C$CLP1
4460 025254                                  FORCEXIT        205$            ;add
4461 025264  005237  002256                  INC     DATA                    ;GET NEXT TEST DATA
4462 025270  023727  002256  000377          CMP     DATA,#377               ;DONE 0 TO 377?
4463 025276  101002                          BHI     205$                    ;BR IF YES
4464 025300  000137  024516                  JMP     100$                    ;DO ANOTHER TEST PATTERN
4465 025304                          205$:
4467 025304                                  ENDSUB                          ;////////////// END SUBTEST //////////
```

```
        025304                                                          L10040:
        025304  104403                                                         TRAP    C$ESUB
  4469  025306  005737  002170                TST     FATFLG                ;ANY FATAL ERRORS ?
  4470  025312  001402                        BEQ     260$                  ;BRANCH IF NOT
  4471  025314  004737  020136                JSR     PC,LKDROP             ;TRY TO DROP THE UNIT
  4472  025320                        260$:
```

```
                                .SBTTL  TEST  1: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

                            ;++
                            ; TEST 1: SUBTEST 3:
                            ;
                            ; SUBTEST DESCRIPTION:
                            ;
                            ;       This subtest verifies the ability of the FIFO to correctly
                            ;       pass a multiple data bytes from input to output.
                            ;       The following sequence is done with various data patterns
                            ;       and byte counts from 2 to 64.
                            ;               1. Initial FIFO status is checked
                            ;               2. The Write FIFO function.
                            ;               3. Read Status is executed and FIFO status is checked.
                            ;               4. Read FIFO is executed and the data and final status
                            ;                  is checked.
                            ;
                            ; TEST STEPS:
                            ;
                            ; BEGIN
                            ;       Write to TSSR to soft initialize
                            ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                            ;       Do a Write Subsystem READ STATUS
                            ;       If Input Ready  NOT=1   Then Print Error
                            ;       If Output Ready NOT=0   Then Print Error
                            ;       If Data In Miss NOT=0   Then Print Error
                            ;       If Last Word    NOT=0   Then Print Error
                            ;  REPEAT FOR DATA 0 TO 377, 377 TO 0, FLOATING 1'S,0'S AND ALL 1'S/0'S
                            ;  REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
                            ;    BEGIN
                            ;       Do a Write Subsystem WRITE NPR to set tape direction out
                            ;       Do a Write Subsystem WRITE FIFO
                            ;       Do a Write Subsystem READ STATUS
                            ;       If Input Ready  NOT=1   Then Print Error
                            ;       If Output Ready NOT=1   Then Print Error
                            ;       If Data In Miss NOT=0   Then Print Error
                            ;       If Last Word    NOT=0   Then Print Error
                            ;       Do Write Subsystem READ FIFO
                            ;       If Data read from FIFO NOT= to Data sent Then Print Error
                            ;       Do a Write Subsystem READ STATUS
                            ;       If Input Ready  NOT=1   Then Print Error
                            ;       If Output Ready NOT=0   Then Print Error
                            ;       If Data In Miss NOT=0   Then Print Error
                            ;       If Last Word    NOT=0   Then Print Error
                            ;    END
                            ;  END
                            ;--
025320                              BGNSUB                          ;////////////// BEGIN SUBTEST //////////
025320                                                                      T1.3:
025320  104402                                                                TRAP    C$BSUB

                            ;       Write to TSSR register to soft initialize the controller
025322                      5$:
025322  004737  016630              JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
025326  103405                      BCS     10$                     ;BR IF SOFT INIT OKAY
025330  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
025332                              ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
```

```
                                                                                    TRAP    C$ERDF
                                                                                    .WORD   116
                                                                                    .WORD   SFIERR
                                                                                    .WORD   SFIMSG
                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                        MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
                        JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
                        BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                        NEXT.ERRNO
                42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                                                                                    TRAP    C$ERDF
                                                                                    .WORD   117
                                                                                    .WORD   T17SSR
                                                                                    .WORD   PKTSSR
                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                    TRAP    C$CLP1
                ;       Do a Write Subsystem READ STATUS
                        JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
                        MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                        BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                        NEXT.ERRNO
                62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                                                                    TRAP    C$ERDF
                                                                                    .WORD   118
                                                                                    .WORD   T173SSR
                                                                                    .WORD   PKTSSR
                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                    TRAP    C$CLP1
                ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
                        JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
                        MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
                        MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
                        BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY=  1
                        BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY=  0
                        BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
                        BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
                ;       If Input Ready NOT=1 then Print Error
                ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
                ;       If Last Word    NOT=0   Then Print Error
                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
                        MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
                        MOV     #T17EXP,R2              ;EXPD ADDRESS
                        MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
                        JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
                        BCS     90$                     ;BR IF YES
                        NEXT.ERRNO
                82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
                                                                                    TRAP    C$ERHRD
```

```
                025530  000167                                                          .WORD   119
                025532  031603                                                          .WORD   T171CMP
                025534  012172                                                          .WORD   MSGSTAT
   4572         025536                          90$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                025536  104406                                                          TRAP    C$CLP1
   4573
   4574
   4575
   4576                                         ;  REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
   4577                                         ;  TSTFLAG =1 FOR INCREMENT TEST PATTERN
   4578                                         ;          =2 FOR DECREMENT TEST PATTERN
   4579                                         ;          =3 FOR TSTBLK TABLE PATTERN
   4580 025540  012737  000001  002260                 MOV     #1,TSTFLAG              ;TEST PATTERN FLAG
   4581 025546                                  95$:
   4582 025546  012737  000002  002254                 MOV     #2,COUNT                ;GET FIRST BYTE COUNT
   4583 025554                                  100$:
   4584                                         ;       Do a Write Subsystem WRITE NPR to set tape direction out
   4585 025554  012700  000100                         MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
   4586 025560  004737  032464                         JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
   4587 025564  012704  033010                         MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
   4588 025570  010465  177776                         MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
   4589 025574  004737  017220                         JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
   4590 025600  103407                                 BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
   4591 025602  010001                                 MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
   4592 025604                                         NEXT.ERRNO
   4593 025604                                  102$:  ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                025604  104455                                                          TRAP    C$ERDF
                025606  000170                                                          .WORD   120
                025610  031431                                                          .WORD   T174SSR
                025612  011670                                                          .WORD   PKTSSR
   4594 025614  004737  020064                         JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
   4595 025620                                  105$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                025620  104406                                                          TRAP    C$CLP1
   4596                                         ;       Do a Write Subsystem WRITE FIFO
   4597 025622  004737  032564                         JSR     PC,T17CLEXP             ;CLEAR EXPD BUFFER
   4598 025626  012701  031114                         MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
   4599 025632  013702  002254                         MOV     COUNT,R2                ;TEST PATTERN SIZE
   4600 025636  022737  000001  002260                 CMP     #1,TSTFLAG              ;INCREMENT PATTERN THIS TIME THRU?
   4601 025644  001005                                 BNE     115$                    ;BR IF NO
   4602 025646  005000                                 CLR     R0                      ;INCREMENT TEST PATTERN
   4603 025650  110021                          110$:  MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
   4604 025652  005200                                 INC     R0                      ;SET NEXT PATTERN
   4605 025654  005302                                 DEC     R2                      ;DONE?
   4606 025656  003374                                 BGT     110$                    ;BR IF NO
   4607 025660  022737  000002  002260  115$:  CMP     #2,TSTFLAG              ;DECREMENT PATTERN THIS TIME THRU?
   4608 025666  001006                                 BNE     125$                    ;BR IF NO
   4609 025670  012700  000377                         MOV     #377,R0                 ;DECREMENT TEST PATTERN
   4610 025674  110021                          120$:  MOVB    R0,(R1)+                ;STORE DECREMENT TEST BYTE
   4611 025676  005300                                 DEC     R0                      ;SET NEXT PATTERN
   4612 025700  005302                                 DEC     R2                      ;DONE?
   4613 025702  003374                                 BGT     120$                    ;BR IF NO
   4614 025704  022737  000003  002260  125$:  CMP     #3,TSTFLAG              ;TSTBLK PATTERNS THIS TIME THRU?
   4615 025712  001005                                 BNE     135$                    ;BR IF NO
   4616 025714  012700  002720                         MOV     #TSTBLK,R0              ;FLOAT 1'S/0'S ETC. TEST TABLE
   4617 025720  112021                          130$:  MOVB    (R0)+,(R1)+             ;STORE A TSTBLK BYTE
   4618 025722  005302                                 DEC     R2                      ;DONE?
   4619 025724  003375                                 BGT     130$                    ;BR IF NO
```

```
 4620 025726                        135$:
 4621 025726  013700  002254              MOV     COUNT,R0                ;FIFO BYTE COUNT
 4622 025732  012701  031114              MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
 4623 025736  004737  032510              JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
 4624 025742  012704  033010              MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4625 025746  010465  177776              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4626 025752  004737  017220              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4627 025756  103407                      BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
 4628 025760  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4629 025762                              NEXT.ERRNO
 4630 025762                        142$: ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      025762  104455                                                                TRAP    C$ERDF
      025764  000171                                                                .WORD   121
      025766  031474                                                                .WORD   T175SSR
      025770  011670                                                                .WORD   PKTSSR
 4631 025772  004737  020064              JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 4632 025776                        150$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      025776  104406                                                                TRAP    C$CLP1
 4633
 4634                               ;     Do a Write Subsystem READ STATUS
 4635 026000  004737  032422              JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
 4636 026004  012704  033010              MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4637 026010  010465  177776              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4638 026014  004737  017220              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4639 026020  103407                      BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
 4640 026022  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4641 026024                              NEXT.ERRNO
 4642 026024                        157$: ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      026024  104455                                                                TRAP    C$ERDF
      026026  000172                                                                .WORD   122
      026030  031364                                                                .WORD   T173SSR
      026032  011670                                                                .WORD   PKTSSR
 4643 026034  004737  020064              JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 4644 026040                        160$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      026040  104406                                                                TRAP    C$CLP1
 4645
 4646                               ;     Set WORDS 0-7 of expd message buffer = to recv since not testing
 4647 026042  004737  032604              JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
 4648 026046  012701  031012              MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 4649 026052  012702  032702              MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 4650 026056  012221                      MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 4651 026060  011211                      MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 4652 026062  052711  000020              BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
 4653 026066  052711  000040              BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
 4654 026072  042711  000200              BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
 4655 026076  042711  000100              BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
 4656                               ;     If Input Ready NOT=1 then Print Error
 4657                               ;     If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
 4658 026102  005000                      CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 4659 026104  012701  032662              MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 4660 026110  012702  030772              MOV     #T17EXP,R2              ;EXPD ADDRESS
 4661 026114  012703  000024              MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 4662 026120  004737  011322              JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 4663 026124  103404                      BCS     170$                    ;BR IF YES
 4664 026126                              NEXT.ERRNO
 4665 026126                        162$: ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
      026126  104456                                                                TRAP    C$ERHRD
```

```
        026130  000173                                                          .WORD   123
        026132  031761                                                          .WORD   T173CMP
        026134  012172                                                          .WORD   MSGSTAT
  4666  026136                    170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        026136  104406                                                          TRAP    C$CLP1
  4667
  4668                            ;       Do Write Subsystem READ FIFO
  4669  026140  013700  002254           MOV     COUNT,R0                ;SET READ BYTE COUNT
  4670  026144  004737  032544           JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
  4671  026150  012704  033010           MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
  4672  026154  010465  177776           MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
  4673  026160  004737  017220           JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  4674  026164  103407                   BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
  4675  026166  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  4676  026170                           NEXT.ERRNO
  4677  026170                    172$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        026170  104455                                                          TRAP    C$ERDF
        026172  000174                                                          .WORD   124
        026174  031540                                                          .WORD   T176SSR
        026176  011670                                                          .WORD   PKTSSR
  4678  026200  004737  020064           JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
  4679  026204                    180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        026204  104406                                                          TRAP    C$CLP1
  4680
  4681                            ;       If Data read from FIFO NOT= to Data sent Then Print Error
  4682  026206  005000                   CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
  4683  026210  012702  031114           MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
  4684  026214  012701  032702           MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
  4685  026220  013703  002254           MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
  4686  026224  004737  011322           JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
  4687  026230  103406                   BCS     200$                    ;BR IF YES
  4688  026232                           NEXT.ERRNO
  4689  026232  013701  002254    192$:  MOV     COUNT,R1                ;GET BYTE COUNT
  4690  026236                           ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
        026236  104456                                                          TRAP    C$ERHRD
        026240  000175                                                          .WORD   125
        026242  032130                                                          .WORD   T175CMP
        026244  012012                                                          .WORD   FIFEXP
  4691  026246                    200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        026246  104406                                                          TRAP    C$CLP1
  4692
  4693                            ;       Do a Write Subsystem READ STATUS
  4694  026250  004737  032422           JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
  4695  026254  012704  033010           MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
  4696  026260  010465  177776           MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
  4697  026264  004737  017220           JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  4698  026270  103407                   BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
  4699  026272  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  4700  026274                           NEXT.ERRNO
  4701  026274                    212$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        026274  104455                                                          TRAP    C$ERDF
        026276  000176                                                          .WORD   126
        026300  031364                                                          .WORD   T173SSR
        026302  011670                                                          .WORD   PKTSSR
  4702  026304  004737  020064           JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
  4703  026310                    220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        026310  104406                                                          TRAP    C$CLP1
```

```
4704                                    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
4705 026312  004737  032604                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
4706 026316  012701  031012                    MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
4707 026322  012702  032702                    MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
4708 026326  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
4709 026330  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
4710 026332  052711  000020                    BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
4711 026336  042711  000040                    BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
4712 026342  042711  000200                    BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
4713 026346  042711  000100                    BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
4714                                    ;       If Input Ready NOT=1 then Print Error
4715                                    ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
4716 026352  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4717 026354  012701  032662                    MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4718 026360  012702  030772                    MOV     #T17EXP,R2              ;EXPD ADDRESS
4719 026364  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4720 026370  004737  011322                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4721 026374  103404                            BCS     240$                    ;BR IF YES
4722 026376                                    NEXT.ERRNO
4723 026376                             232$:  ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     026376  104456                                                                    TRAP    C$ERHRD
     026400  000177                                                                    .WORD   127
     026402  032045                                                                    .WORD   T174CMP
     026404  012172                                                                    .WORD   MSGSTAT
4724 026406                             240$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     026406  104406                                                                    TRAP    C$CLP1
4725 026410                                    FORCEXIT        250$            ;@@D
4726 026420  005237  002254                    INC     COUNT                   ;GET NEXT BYTE COUNT
4727 026424  023727  002254  000077            CMP     COUNT,#77               ;DONE 0 TO 77
4728 026432  101002                            BHI     250$                    ;BR IF YES
4729 026434  000137  025554                    JMP     100$                    ;DO ANOTHER BYTE COUNT
4730 026440  005237  002260             250$:  INC     TSTFLAG                 ;GET NEXT TEST PATTERN CODE
4731 026444  023727  002260  000003            CMP     TSTFLAG,#3              ;DONE INC,DEC,TSTBLK PATTERNS?
4732 026452  101002                            BHI     255$                    ;BR IF YES
4733 026454  000137  025546                    JMP     95$                     ;DO ANOTHER TEST PATTERN
4734 026460                             255$:
4735 026460                                    ENDSUB                          ;////////////// END SUBTEST //////////
     026460                                                                             L10041:
     026460  104403                                                                    TRAP    C$ESUB
4736
4737 026462  005737  002170                    TST     FATFLG                  ;ANY FATAL ERRORS ?
4738 026466  001402                            BEQ     260$                    ;BRANCH IF NOT
4739 026470  004737  020136                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4740 026474                             260$:
4741
4742
4743
```

```
                                        .SBTTL  TEST  1: SUBTEST 4; FIFO Verify ILW Status

                                ;++
                                ; TEST 1: SUBTEST 4;
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies that reading the FIFO when it is
                                ;       empty causes the Last Word (ILW) status to assert.
                                ;
                                ; TEST STEPS:
                                ;
                                ; BEGIN
                                ;       Write to TSSR to soft initialize
                                ;       Do Write Subsystem READ FIFO with byte count equal to 1
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Input Ready  NOT=1   Then Print Error
                                ;       If Output Ready NOT=0   Then Print Error
                                ;       If Data In Miss NOT=0   Then Print Error
                                ;       If Last Word (ILW) NOT=1 Then Print Error
                                ;   END
                                ;--
026474                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                        T1.4:
026474  104402                                                                          TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
026476                          5$:
026476  004737  016630          JSR     PC,SOFINIT                      ;WRITE TO TSSR TO SOFT INITIALIZE
026502  103405                  BCS     10$                             ;BR IF SOFT INIT OKAY
026504  010001                  MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
026506                          ERRDF   ERRNO,SFIERR,SFIMSG             ;DEVICE FATAL DURING INIT
026506  104455                                                                          TRAP    C$ERDF
026510  000177                                                                        .WORD   127
026512  003550                                                                        .WORD   SFIERR
026514  011656                                                                        .WORD   SFIMSG

                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
026516  005037  002170          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
026522  012704  032640          MOV     #T17PACKET,R4                   ;GET THE ADDRESS OF COMMAND PACKET
026526  004737  010322          JSR     PC,WRTCHR                       ;DO WRITE CHARACTERISTICS COMMAND
026532  103407                  BCS     50$                             ;BR IF CARRY SET (GOOD RETURN)
026534  010001                  MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
026536                          NEXT.ERRNO
026536                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
026536  104455                                                                          TRAP    C$ERDF
026540  000200                                                                        .WORD   128
026542  031263                                                                        .WORD   T17SSR
026544  011670                                                                        .WORD   PKTSSR
026546  004737  020064          JSR     PC,FATCHK                       ;INC AND CHECK FOR MORE THAN 25 ERRORS
026552                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
026552  104406                                                                          TRAP    C$CLP1

                                ;       Do Write Subsystem READ FIFO with byte count equal to 1
026554  012700  000001          MOV     #1,R0                           ;SET READ BYTE COUNT
026560  004737  032544          JSR     PC,T17RFIF                      ;SETUP T17PK2 FOR READ FIFO
026564  012704  033010          MOV     #T17PK2,R4                      ;GET WRITE SUBSYSTEM COMMAND PACKET
026570  010465  177776          MOV     R4,TSDB(R5)                     ;SET THE PACKET ADDRESS TO EXECUTE
```

```
4791 026574  004737  017220                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4792 026600  103407                        BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
4793 026602  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4794 026604                                NEXT.ERRNO
4795 026604                        142$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     026604  104455                                                            TRAP    C$ERDF
     026606  000201                                                            .WORD   129
     026610  031540                                                            .WORD   T176SSR
     026612  011670                                                            .WORD   PKTSSR
4796 026614  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4797 026620                        150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     026620  104406                                                            TRAP    C$CLP1
4798
4799                               ;       Do a Write Subsystem READ STATUS
4800 026622  004737  032422                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
4801 026626  012704  033010                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4802 026632  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4803 026636  004737  017220                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4804 026642  103407                        BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
4805 026644  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4806 026646                                NEXT.ERRNO
4807 026646                        162$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     026646  104455                                                            TRAP    C$ERDF
     026650  000202                                                            .WORD   130
     026652  031364                                                            .WORD   T173SSR
     026654  011670                                                            .WORD   PKTSSR
4808 026656  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4809 026662                        170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     026662  104406                                                            TRAP    C$CLP1
4810                               ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
4811 026664  004737  032604                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
4812 026670  012701  031012                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
4813 026674  012702  032702                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
4814 026700  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
4815 026702  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
4816 026704  052711  000020                BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
4817 026710  042711  000040                BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
4818 026714  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
4819 026720  052711  000100                BIS     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=1
4820                               ;       If Input Ready NOT=1 then Print Error
4821                               ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
4822                               ;       If Last Word (ILW) NOT=1 Then Print Error
4823 026724  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4824 026726  012701  032662                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4825 026732  012702  030772                MOV     #T17EXP,R2              ;EXPD ADDRESS
4826 026736  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4827 026742  004737  011322                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4828 026746  103404                        BCS     180$                    ;BR IF YES
4829 026750                                NEXT.ERRNO
4830 026750                        172$:   ERRHRD  ERRNO,T176CMP,MSGSTAT   ;REPORT ERROR
     026750  104456                                                            TRAP    C$ERHRD
     026752  000203                                                            .WORD   131
     026754  032204                                                            .WORD   T176CMP
     026756  012172                                                            .WORD   MSGSTAT
4831 026760                        180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     026760  104406                                                            TRAP    C$CLP1
4832
```

```
4833 026762                        ENDSUB                          ;///////////// END SUBTEST //////////
     026762                                                                      L10042:
     026762  104403                                                                       TRAP    C$ESUB
4834
4835 026764  005737  002170        TST     FATFLG                  ;ANY FATAL ERRORS ?
4836 026770  001402                BEQ     260$                    ;BRANCH IF NOT
4837 026772  004737  020136        JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4838 026776                 260$:
4839
4840
```

```
                                        .SBTTL  TEST  1: SUBTEST 5: FIFO Verify Input Ready

                                ;++
                                ; TEST 1: SUBTEST 5;
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies that writing 64. bytes into the FIFO
                                ;       without reading any out causes the Input Ready status to
                                ;       negate. The Subtest then verifies that writing a 65th byte
                                ;       into the FIFO causes the Data In Miss status to assert.
                                ;       Next it is verified that the original 64 bytes can be read
                                ;       out correctly and that the data has not been corrupted.
                                ;
                                ; TEST STEPS:
                                ;
                                ; BEGIN
                                ;       Write to TSSR to soft initialize
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                                ;       Do a Write Subsystem WRITE NPR to set tape direction out
                                ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Input Ready  NOT=0   Then Print Error
                                ;       If Output Ready NOT=1   Then Print Error
                                ;       If Data In Miss NOT=0   Then Print Error
                                ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Input Ready  NOT=0   Then Print Error
                                ;       If Output Ready NOT=1   Then Print Error
                                ;       If Data In Miss NOT=1   Then Print Error
                                ;       Do Write Subsystem READ FIFO
                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Input Ready  NOT=1   Then Print Error
                                ;       If Output Ready NOT=0   Then Print Error
                                ;       If Data In Miss NOT=1   Then Print Error
                                ; END
                                ;--
026776                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
026776                                                                  T1.5:
026776  104402                                                          TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
027000                          5$:
027000  004737  016630          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
027004  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
027006  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027010                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
027010  104455                                                          TRAP    C$ERDF
027012  000203                                                          .WORD   131
027014  003550                                                          .WORD   SFIERR
027016  011656                                                          .WORD   SFIMSG

                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
027020  005037  002170          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
027024  012704  032640          MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
027030  004737  010322          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
027034  103407                  BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
```

```
4893 027036  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4894 027040                                  NEXT.ERRNO
4895 027040                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     027040  104455                                                                  TRAP    C$ERDF
     027042  000204                                                                  .WORD   132
     027044  031263                                                                  .WORD   T17SSR
     027046  011670                                                                  .WORD   PKTSSR
4896 027050  004737  020064                     JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4897 027054                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027054  104406                                                                  TRAP    C$CLP1
4898
4899                                 ;       Do a Write Subsystem WRITE NPR to set tape direction out
4900 027056  012700  000100          100$:   MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
4901 027062  004737  032464                     JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
4902 027066  012704  033010                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4903 027072  010465  177776                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4904 027076  004737  017220                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4905 027102  103407                             BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
4906 027104  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4907 027106                                  NEXT.ERRNO
4908 027106                          102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     027106  104455                                                                  TRAP    C$ERDF
     027110  000205                                                                  .WORD   133
     027112  031431                                                                  .WORD   T174SSR
     027114  011670                                                                  .WORD   PKTSSR
4909 027116  004737  020064                     JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4910 027122                          105$:   CKLOOP                          ;LUOP ON ERROR, IF FLAG SET
     027122  104406                                                                  TRAP    C$CLP1
4911
4912                                 ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
4913 027124  012737  000100  002254          MOV     #64.,COUNT              ;WRITE 64 BYTES
4914 027132  012701  031114                     MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
4915 027136  012702  000100                     MOV     #64.,R2                 ;TEST PATTERN SIZE
4916 027142  005000                             CLR     R0                      ;INCREMENT TEST PATTERN
4917 027144  110021                  110$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
4918 027146  005200                             INC     R0                      ;SET NEXT PATTERN
4919 027150  005302                             DEC     R2                      ;DONE?
4920 027152  003374                             BGT     110$                    ;BR IF NO
4921 027154  013700  002254                     MOV     COUNT,R0                ;FIFO BYTE COUNT
4922 027160  012701  031114                     MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
4923 027164  004737  032510                     JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
4924 027170  012704  033010                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4925 027174  010465  177776                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4926 027200  004737  017220                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4927 027204  103407                             BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
4928 027206  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4929 027210                                  NEXT.ERRNO
4930 027210                          142$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     027210  104455                                                                  TRAP    C$ERDF
     027212  000206                                                                  .WORD   134
     027214  031474                                                                  .WORD   T175SSR
     027216  011670                                                                  .WORD   PKTSSR
4931 027220  004737  020064                     JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4932 027224                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027224  104406                                                                  TRAP    C$CLP1
4933
4934                                 ;       Do a Write Subsystem READ STATUS
```

```
4935                                    ;       If Input Ready  NOT=0   Then Print Error
4936                                    ;       If Output Ready NOT=1   Then Print Error
4937                                    ;       If Data In Miss NOT=0   Then Print Error
4938 027226  004737  032422                     JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
4939 027232  012704  033010                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4940 027236  010465  177776                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4941 027242  004737  017220                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4942 027246  103407                             BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
4943 027250  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4944 027252                                     NEXT.ERRNO
4945 027252                             157$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     027252  104455                                                                     TRAP    C$ERDF
     027254  000207                                                                     .WORD   135
     027256  031364                                                                     .WORD   T173SSR
     027260  011670                                                                     .WORD   PKTSSR
4946 027262  004737  020064                     JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4947 027266                             160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027266  104406                                                                     TRAP    C$CLP1
4948                                    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
4949 027270  004737  032604                     JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
4950 027274  012701  031012                     MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
4951 027300  012702  032702                     MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
4952 027304  012221                             MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
4953 027306  011211                             MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
4954 027310  042711  000020                     BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
4955 027314  052711  000040                     BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
4956 027320  042711  000200                     BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
4957 027324  005000                             CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4958 027326  012701  032662                     MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4959 027332  012702  030772                     MOV     #T17EXP,R2              ;EXPD ADDRESS
4960 027336  012703  000024                     MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4961 027342  004737  011322                     JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4962 027346  103404                             BCS     170$                    ;BR IF YES
4963 027350                                     NEXT.ERRNO
4964 027350                             162$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     027350  104456                                                                     TRAP    C$ERHRD
     027352  000210                                                                     .WORD   136
     027354  031761                                                                     .WORD   T173CMP
     027356  012172                                                                     .WORD   MSGSTAT
4965 027360                             170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027360  104406                                                                     TRAP    C$CLP1
4966
4967
4968                                    ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
4969 027362  012700  000001                     MOV     #1,R0                   ;FIFO BYTE COUNT
4970 027366  012701  031114                     MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
4971 027372  004737  032510                     JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
4972 027376  012704  033010                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4973 027402  010465  177776                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4974 027406  004737  017220                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4975 027412  103407                             BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
4976 027414  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4977 027416                                     NEXT.ERRNO
4978 027416                             172$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     027416  104455                                                                     TRAP    C$ERDF
     027420  000211                                                                     .WORD   137
     027422  031474                                                                     .WORD   T175SSR
```

```
     027424  011670                                                          .WORD   PKTSSR
4979 027426  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4980 027432                        180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027432  104406                                                          TRAP    C$CLP1
4981
4982                               ;       Do a Write Subsystem READ STATUS
4983                               ;       If Input Ready  NOT=0   Then Print Error
4984                               ;       If Output Ready NOT=1   Then Print Error
4985                               ;       If Data In Miss NOT=1   Then Print Error
4986 027434  004737  032422                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
4987 027440  012704  033010                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4988 027444  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4989 027450  004737  017220                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4990 027454  103407                        BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
4991 027456  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4992 027460                                NEXT.ERRNO
4993 027460                        187$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     027460  104455                                                          TRAP    C$ERDF
     027462  000212                                                          .WORD   138
     027464  031364                                                          .WORD   T173SSR
     027466  011670                                                          .WORD   PKTSSR
4994 027470  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
4995 027474                        190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027474  104406                                                          TRAP    C$CLP1
4996                               ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
4997 027476  004737  032604                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
4998 027502  012701  031012                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
4999 027506  012702  032702                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
5000 027512  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
5001 027514  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
5002 027516  042711  000020                BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
5003 027522  052711  000040                BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
5004 027526  052711  000200                BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
5005 027532  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
5006 027534  012701  032662                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
5007 027540  012702  030772                MOV     #T17EXP,R2              ;EXPD ADDRESS
5008 027544  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
5009 027550  004737  011322                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
5010 027554  103404                        BCS     200$                    ;BR IF YES
5011 027556                                NEXT.ERRNO
5012 027556                        192$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     027556  104456                                                          TRAP    C$ERHRD
     027560  000213                                                          .WORD   139
     027562  031761                                                          .WORD   T173CMP
     027564  012172                                                          .WORD   MSGSTAT
5013 027566                        200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027566  104406                                                          TRAP    C$CLP1
5014                               ;       Do Write Subsystem READ FIFO
5015 027570  013700  002254                MOV     COUNT,R0                ;SET READ BYTE COUNT
5016 027574  004737  032544                JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
5017 027600  012704  033010                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5018 027604  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5019 027610  004737  017220                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5020 027614  103407                        BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
5021 027616  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5022 027620                                NEXT.ERRNO
5023 027620                        212$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
```

```
                027620  104455                                                          TRAP    C$ERDF
                027622  000214                                                          .WORD   140
                027624  031540                                                          .WORD   T176SSR
                027626  011670                                                          .WORD   PKTSSR
     5024       027630  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
     5025       027634                          220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                027634  104406                                                          TRAP    C$CLP1
     5026
     5027                                       ;       If Data read from FIFO NOT= to Data sent Then Print Error
     5028       027636  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
     5029       027640  012702  031114                  MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
     5030       027644  012701  032702                  MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
     5031       027650  013703  002254                  MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
     5032       027654  004737  011322                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
     5033       027660  103406                          BCS     240$                    ;BR IF YES
     5034       027662                                  NEXT.ERRNO
     5035       027662  013701  002254          232$:   MOV     COUNT,R1                ;GET BYTE COUNT
     5036       027666                                  ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
                027666  104456                                                          TRAP    C$ERHRD
                027670  000215                                                          .WORD   141
                027672  032130                                                          .WORD   T175CMP
                027674  012012                                                          .WORD   FIFEXP
     5037       027676                          240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                027676  104406                                                          TRAP    C$CLP1
     5038
     5039                                       ;       Do a Write Subsystem READ STATUS
     5040                                       ;       If Input Ready  NOT=1   Then Print Error
     5041                                       ;       If Output Ready NOT=0   Then Print Error
     5042                                       ;       If Data In Miss NOT=1   Then Print Error
     5043       027700  004737  032422                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
     5044       027704  012704  033010                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
     5045       027710  010465  177776                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
     5046       027714  004737  017220                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
     5047       027720  103407                          BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
     5048       027722  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     5049       027724                                  NEXT.ERRNO
     5050       027724                          252$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                027724  104455                                                          TRAP    C$ERDF
                027726  000216                                                          .WORD   142
                027730  031364                                                          .WORD   T173SSR
                027732  011670                                                          .WORD   PKTSSR
     5051       027734  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
     5052       027740                          260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                027740  104406                                                          TRAP    C$CLP1
     5053                                       ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
     5054       027742  004737  032604                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
     5055       027746  012701  031012                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
     5056       027752  012702  032702                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
     5057       027756  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
     5058       027760  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
     5059       027762  052711  000020                  BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
     5060       027766  042711  000040                  BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
     5061       027772  052711  000200                  BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
     5062       027776  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
     5063       030000  012701  032662                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
     5064       030004  012702  030772                  MOV     #T17EXP,R2              ;EXPD ADDRESS
     5065       030010  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
```

```
 5066 030014  004737  011322                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 5067 030020  103404                        BCS     280$                    ;BR IF YES
 5068 030022                                NEXT.ERRNO
 5069 030022                        272$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
      030022  104456                                                                TRAP    C$ERHRD
      030024  000217                                                                .WORD   143
      030026  032045                                                                .WORD   T174CMP
      030030  012172                                                                .WORD   MSGSTAT
 5070 030032                        280$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      030032  104406                                                                TRAP    C$CLP1
 5071
 5072 030034                                ENDSUB                          ;////////////// END SUBTEST //////////
      030034                                                                      L10043:
      030034  104403                                                                TRAP    C$ESUB
 5073
 5074 030036  005737  002170                TST     FATFLG                  ;ANY FATAL ERRORS ?
 5075 030042  001402                        BEQ     300$                    ;BRANCH IF NOT
 5076 030044  004737  020136                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 5077 030050                        300$:
 5078
 5079
 5080
```

```
                                .SBTTL  TEST  1: SUBTEST 6; FIFO Verify Reset FIFO Test

                              ;++
                              ; TEST 1: SUBTEST 6;
                              ;
                              ; SUBTEST DESCRIPTION:
                              ;
                              ;       This subtest verifies that the Reset FIFO function within
                              ;       the Write Miscellaneous Control 1 function initializes
                              ;       the FIFO to correct intial status. The following steps
                              ;       are performed:
                              ;               1. Reset an already initialized FIFO and check for
                              ;                  proper status.
                              ;               2. Write a varying number of bytes (1-65.) into the
                              ;                  FIFO and verify that after each block of bytes is
                              ;                  written the FIFO can be be reset to it's initial
                              ;                  state.
                              ;
                              ; TEST STEPS:
                              ;
                              ; BEGIN
                              ;       Write to TSSR to soft initialize
                              ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                              ;       Do a Write Subsystem Write Misc to Reset FIFO
                              ;       Do a Write Subsystem READ STATUS
                              ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
                              ;         signals NOT=0 Then Print Error
                              ;       Do a Write Subsystem WRITE NPR to set tape direction out
                              ;
                              ; REPEAT FOR BYTE COUNT 1 TO 65.
                              ;   BEGIN
                              ;      Do a Write Subsystem WRITE FIFO with the current byte count
                              ;      Do a Write Subsystem Write Misc to Reset FIFO
                              ;      Do a Write Subsystem READ STATUS
                              ;      If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
                              ;        signals NOT=0 Then Print Error
                              ;   END
                              ;--
030050                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////
030050                                                                                  T1.6:
030050  104402                                                                                TRAP    C$BSUB

                              ;       Write to TSSR register to soft initialize the controller
                              5$:
030052  004737  016630                JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
030056  103405                        BCS     10$                     ;BR IF SOFT INIT OKAY
030060  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
030062                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
030062  104455                                                                            TRAP    C$ERDF
030064  000217                                                                          .WORD   143
030066  003550                                                                          .WORD   SFIERR
030070  011656                                                                          .WORD   SFIMSG

                              ;       Do a WRITE CHARACTERISTICS to setup a message buffer
030072  005037  002170        10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
030076  012704  032640                MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
030102  004737  010322                JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
030106  103407                        BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
```

```
 5133 030110  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5134 030112                                NEXT.ERRNO
 5135 030112                        42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      030112  104455                                                         TRAP    C$ERDF
      030114  000220                                                         .WORD   144
      030116  031263                                                         .WORD   T17SSR
      030120  011670                                                         .WORD   PKTSSR
 5136 030122  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5137 030126                        50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      030126  104406                                                         TRAP    C$CLP1
 5138                               ;       Do a Write Subsystem Write Misc to Reset FIFO
 5139 030130  004737  032442                JSR     PC,T17RSFIF             ;SETUP PKT FOR WRITE MISC RESET FIFO
 5140 030134  012704  033010                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 5141 030140  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 5142 030144  004737  017220                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 5143 030150  103407                        BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
 5144 030152  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5145 030154                                NEXT.ERRNO
 5146 030154                        62$:    ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      030154  104455                                                         TRAP    C$ERDF
      030156  000221                                                         .WORD   145
      030160  031320                                                         .WORD   T172SSR
      030162  011670                                                         .WORD   PKTSSR
 5147 030164  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5148 030170                        70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      030170  104406                                                         TRAP    C$CLP1
 5149
 5150                               ;       Do a Write Subsystem READ STATUS
 5151                               ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
 5152                               ;         signals NOT=0 Then Print Error
 5153 030172  004737  032422                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
 5154 030176  012704  033010                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 5155 030202  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 5156 030206  004737  017220                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 5157 030212  103407                        BCS     80$                     ;BR IF CARRY SET (GOOD RETURN)
 5158 030214  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5159 030216                                NEXT.ERRNO
 5160 030216                        77$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      030216  104455                                                         TRAP    C$ERDF
      030220  000222                                                         .WORD   146
      030222  031364                                                         .WORD   T173SSR
      030224  011670                                                         .WORD   PKTSSR
 5161 030226  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5162 030232                        80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      030232  104406                                                         TRAP    C$CLP1
 5163 030234  004737  032604                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 5164 030240  012701  031012                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 5165 030244  012702  032702                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 5166 030250  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
 5167 030252  042711  002000                BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
 5168 030256  042711  001000                BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
 5169 030262  042711  000400                BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
 5170 030266  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
 5171 030274  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 5172 030276  012701  032662                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 5173 030302  012702  030772                MOV     #T17EXP,R2              ;EXPD ADDRESS
 5174 030306  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
```

```
 5175 030312  004737  011322                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 5176 030316  103404                        BCS     100$                    ;BR IF YES
 5177 030320                                NEXT.ERRNO
 5178 030320                          92$:  ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
      030320  104456                                                        TRAP    C$ERHRD
      030322  000223                                                        .WORD   147
      030324  032312                                                        .WORD   T177CMP
      030326  012172                                                        .WORD   MSGSTAT
 5179 030330                          100$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      030330  104406                                                        TRAP    C$CLP1
 5180
 5181                                 ;     Do a Write Subsystem WRITE NPR to set tape direction out
 5182 030332  012700  000100                MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
 5183 030336  004737  032464                JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
 5184 030342  012704  033010                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 5185 030346  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 5186 030352  004737  017220                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 5187 030356  103407                        BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
 5188 030360  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5189 030362                                NEXT.ERRNO
 5190 030362                          112$: ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      030362  104455                                                        TRAP    C$ERDF
      030364  000224                                                        .WORD   148
      030366  031431                                                        .WORD   T174SSR
      030370  011670                                                        .WORD   PKTSSR
 5191 030372  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5192 030376                          120$: CKLOOP                          ;LUOP ON ERROR, IF FLAG SET
      030376  104406                                                        TRAP    C$CLP1
 5193
 5194                                 ;     Setup incrementing pattern in FIFO data buffer
 5195 030400  012701  031012                MOV     #T17EXSTA,R1            ;EXPD WRITE FIFO DATA BUFFER
 5196 030404  012702  000100                MOV     #64.,R2                 ;TEST PATTERN SIZE
 5197 030410  005000                        CLR     R0                      ;INCREMENT TEST PATTERN
 5198 030412  110021                  130$: MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
 5199 030414  005200                        INC     R0                      ;SET NEXT PATTERN
 5200 030416  005302                        DEC     R2                      ;DONE?
 5201 030420  003374                        BGT     130$                    ;BR IF NO
 5202
 5203                                 ;  REPEAT FOR BYTE COUNT 1 TO 65.
 5204 030422  012737  000001  002254        MOV     #1,COUNT                ;GET FIRST BYTE COUNT
 5205                                 ;     Do a Write Subsystem WRITE FIFO with the current byte count
 5206 030430                          150$:                                 ;REPEAT LOOP LABEL
 5207 030430  013700  002254                MOV     COUNT,R0                ;FIFO BYTE COUNT
 5208 030434  012701  031012                MOV     #T17EXSTA,R1            ;FIFO WRITE DATA ADDRESS
 5209 030440  004737  032510                JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
 5210 030444  012704  033010                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 5211 030450  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 5212 030454  004737  017220                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 5213 030460  103407                        BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
 5214 030462  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5215 030464                                NEXT.ERRNO
 5216 030464                          152$: ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      030464  104455                                                        TRAP    C$ERDF
      030466  000225                                                        .WORD   149
      030470  031474                                                        .WORD   T175SSR
      030472  011670                                                        .WORD   PKTSSR
 5217 030474  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
```

```
5218 030500                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     030500  104406                                                                      TRAP    C$CLP1
5219
5220                                 ;       Do a Write Subsystem Write Misc to Reset FIFO
5221 030502  004737  032442                  JSR     PC,I17RSFIF             ;SETUP PKT FOR WRITE MISC RESET FIFO
5222 030506  012704  033010                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5223 030512  010465  177776                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5224 030516  004737  017220                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5225 030522  103407                          BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
5226 030524  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5227 030526                                  NEXT.ERRNO
5228 030526                          162$:   ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     030526  104455                                                                      TRAP    C$ERDF
     030530  000226                                                                      .WORD   150
     030532  031320                                                                      .WORD   T172SSR
     030534  011670                                                                      .WORD   PKTSSR
5229 030536  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5230 030542                          170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     030542  104406                                                                      TRAP    C$CLP1
5231
5232                                 ;       Do a Write Subsystem READ STATUS
5233                                 ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
5234                                 ;         signals NOT=0 Then Print Error
5235 030544  004737  032422                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
5236 030550  012704  033010                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5237 030554  010465  177776                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5238 030560  004737  017220                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5239 030564  103407                          BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
5240 030566  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5241 030570                                  NEXT.ERRNO
5242 030570                          177$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     030570  104455                                                                      TRAP    C$ERDF
     030572  000227                                                                      .WORD   151
     030574  031364                                                                      .WORD   T173SSR
     030576  011670                                                                      .WORD   PKTSSR
5243 030600  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5244 030604                          180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     030604  104406                                                                      TRAP    C$CLP1
5245 030606  004737  032604                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5246 030612  012701  031012                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
5247 030616  012702  032702                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
5248 030622  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
5249 030624  042711  002000                  BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
5250 030630  042711  001000                  BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
5251 030634  042711  000400                  BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
5252 030640  016261  000002  000002          MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
5253 030646  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
5254 030650  012701  032662                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
5255 030654  012702  030772                  MOV     #T17EXP,R2              ;EXPD ADDRESS
5256 030660  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
5257 030664  004737  011322                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
5258 030670  103404                          BCS     200$                    ;BR IF YES
5259 030672                                  NEXT.ERRNO
5260 030672                          192$:   ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
     030672  104456                                                                      TRAP    C$ERHRD
     030674  000230                                                                      .WORD   152
     030676  032312                                                                      .WORD   T177CMP
```

```
                030700  012172                                                          .WORD   MSGSTAT
5261            030702                  200$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                030702  104406                                                          TRAP    C$CLP1
5262
5263
5264            030704                  250$:
5265            030704                          FORCEXIT        260$
5266            030714  005237  002254          INC     COUNT           ;GET NEXT BYTE COUNT
5267            030720  023727  002254  000101  CMP     COUNT,#65.      ;DONE ALL BYTES?
5268            030726  101002          BHI     260$                    ;BR IF YES
5269            030730  000137  030430          JMP     150$            ;DO ANOTHER BYTE COUNT
5270            030734                  260$:
5271
5272            030734                          ENDSUB                  ;////////////// END SUBTEST //////////
                030734                                                                  L10044:
                030734  104403                                                          TRAP    C$ESUB
5273
5274            030736  005737  002170          TST     FATFLG          ;ANY FATAL ERRORS ?
5275            030742  001402          BEQ     300$                    ;BRANCH IF NOT
5276            030744  004737  020136          JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
5277            030750  004737  017340  300$:   JSR     PC,TSTLOOP      ;DO ITERATIONS?
5278            030754  103002          BCC     305$                    ;BR IF NO
5279            030756  000137  024046          JMP     T17LOOP         ;LOOP UNTIL ITERATIONS DONE
5280            030762                  305$:
5281            030762                  TMPEND:                 ;aaa TEMPORARY END OF TEST
5282
5283            030762                          EXIT    TST             ;////////// EXIT TEST //////////
                030762  104432                                                          TRAP    C$EXIT
                030764  002144                                                          .WORD   L10036-.
5284
5285
```

```
                                        ;+
                                        ;LOCAL STORAGE FOR THIS TEST
                                        ;-


030766                                  T17MSK:                         ;MASK OF UNTESTED BITS IN READ STATUS BYTES
                                                                        ;UNTESTED BITS ARE SET TO 1
030766      377                                 .BYTE   ^C<000>         ;BYTE 0 MASK
030767      037                                 .BYTE   ^C<340>         ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
030770      360                                 .BYTE   ^C<017>         ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
030771      000                                 .BYTE   0               ;MAKE IT EVEN

030772                                  T17EXP:                         ;BEGIN EXPECTED DATA BUFFER
030772  000000                              .WORD   0                   ;MESSAGE TYPE
030774  000000                              .WORD   0                   ;DATA FIELD LENGTH
030776  000000                              .WORD   0                   ;RBPCR
031000  000000                              .WORD   0                   ;XST0
031002  000000                              .WORD   0                   ;XST1
031004  000000                              .WORD   0                   ;XST2
031006  000000                              .WORD   0                   ;XST3
031010  000000                              .WORD   0                   ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
031012                                  T17EXSTA: .BLKB 66.             ;EXPECTED READ STATUS AND WRITE FIFO DATA
031114                                  T17EXEND:                               ;END EXPECTED DATA BUFFER

031114                                  T17WFDATA: .BLKB 66.            ;WRITE FIFO EXPECTED DATA BUFFER

                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

031216      106     111     106     TST17ID:        .ASCIZ  'FIFO Exerciser'

031235      122     105     127     T17RWN: .ASCIZ  'REWIND Command Failed'
031263      127     122     111     T17SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
031320      127     122     111     T172SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
031364      127     122     111     T173SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
031431      127     122     111     T174SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
031474      127     122     111     T175SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
031540      127     122     111     T176SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
031603      106     111     106     T171CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize'
031665      122     145     141     T172CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
031761      106     111     106     T173CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after WRITE FIFO'
032045      106     111     106     T174CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO'
032130      122     145     141     T175CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
032204      106     111     106     T176CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO from an Empty FIFO'
032312      106     111     106     T177CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after RESET FIFO'
                                            .EVEN

                                        ;+
                                        ; CLEAR MESSAGE BUFFER
                                        ;-
032376                                  T17CLRBUF:                      ;
032376                                      SAVREG                      ;SAVE R1-R5 UNTIL NEXT RETURN
032402  012701  032662                      MOV     #T17BFR,R1          ;GET MESSAGE BUFFER ADDRESS
032406  012702  000120                      MOV     #T17BEND-T17BFR,R2  ;SIZE OF MESSAGE BUFFER IN BYTES
032412  105021                      10$:    CLRB    (R1)+               ;CLEAR A BYTE
032414  005302                              DEC     R2                  ;DONE?
```

```
 5344 032416  003375                        BGT     10$                     ;BR IF NO
 5345 032420  000207                        RTS     PC                      ;RETURN
 5346
 5347                                ;+
 5348                                ; SETUP T17PK2 PACKET FOR READ STATUS
 5349                                ;-
 5350 032422                        T17SRD:
 5351 032422  004737  032376                JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
 5352 032426  012700  033020                MOV     #T17DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
 5353 032432  112720  000005                MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
 5354 032436  105010                        CLRB    (R0)                    ;CLEAR BSEL1
 5355 032440  000207                        RTS     PC                      ;RETURN
 5356
 5357                                ;+
 5358                                ; SETUP T17PK2 PACKET FOR WRITE MISC RESET FIFO
 5359                                ;-
 5360 032442                        T17RSFIF:
 5361 032442  004737  032376                JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
 5362 032446  012700  033020                MOV     #T17DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
 5363 032452  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
 5364 032456  112710  000030                MOVB    #MS.RSFIF!MS.RSTAP,(R0) ;STORE BSEL1 CLEAR FIFO CODES
 5365 032462  000207                        RTS     PC                      ;RETURN
 5366
 5367                                ;+
 5368                                ; SETUP T17PK2 PACKET FOR WRITE NPR
 5369                                ;
 5370                                ; INPUT:
 5371                                ;       R0 CONTAINS BSEL1 NPR DATA
 5372                                ;
 5373                                ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
 5374                                ;-
 5375 032464                        T17SNPR:
 5376 032464  004737  032376                JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
 5377 032470  012701  033020                MOV     #T17DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
 5378 032474  112721  000011                MOVB    #PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
 5379 032500  052700  000020                BIS     #NP.WRP,R0              ;DON'T WRITE WRONG PARITY
 5380 032504  110011                        MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
 5381 032506  000207                        RTS     PC                      ;RETURN
 5382
 5383                                ;+
 5384                                ; SETUP T17PK2 PACKET FOR WRITE FIFO
 5385                                ;
 5386                                ; INPUT:
 5387                                ;       R0 CONTAINS BYTE COUNT
 5388                                ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
 5389                                ;-
 5390 032510                        T17WFIF:
 5391 032510                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
 5392 032514  004737  032376                JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
 5393 032520  012702  033020                MOV     #T17DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
 5394 032524  112722  000004                MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
 5395 032530  110022                        MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
 5396 032532  005022                        CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
 5397 032534  112122                10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
 5398 032536  005300                        DEC     R0                      ;DONE ALL BYTES?
 5399 032540  003375                        BGT     10$                     ;BR IF NO
 5400 032542  000207                        RTS     PC                      ;RETURN
```

```
                                        ;+
                                        ; SETUP T17PK2 PACKET FOR READ FIFO
                                        ;
                                        ; INPUT:
                                        ;       R0 CONTAINS SEL2 BYTE COUNT
                                        ;-
032544                                  T17RFIF:
032544  004737  032376                          JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
032550  012701  033020                          MOV     #T17DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
032554  112721  000003                          MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
032560  110021                                  MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
032562  000207                                  RTS     PC                      ;RETURN
                                        ;+
                                        ; CLEAR EXPECTED DATA MESSAGE BUFFER
                                        ;-
032564                                  T17CLEXP:
032564  012701  030772                          MOV     #T17EXP,R1              ;GET EXPD ADDRESS
032570  012700  000122                          MOV     #T17EXEND-T17EXP,R0     ;GET EXPD SIZE
032574  105021                          10$:    CLRB    (R1)+                   ;CLEAR A BYTE
032576  005300                                  DEC     R0                      ;DONE?
032600  003375                                  BGT     10$                     ;BR IF NO
032602  000207                                  RTS     PC                      ;RETURN

                                        ;+
                                        ;Set WORDS 0-7 of expd message buffer = to recv since not testing
                                        ;-
032604                                  T17SETEXP:
032604  012702  030772                          MOV     #T17EXP,R2              ;GET EXPD
032610  012703  032662                          MOV     #T17BFR,R3              ;GET READ STATUS RECV BUFFER
032614  012700  000010                          MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
032620  012322                          5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
032622  005300                                  DEC     R0                      ;DONE WORDS 0-7 WORDS?
032624  003375                                  BGT     5$                      ;BR IF NO
032626  000207                                  RTS     PC                      ;RETURN

032630                                          .BLKB   10-<.-TUV2A87>
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND PACKET
                                        ;
032640                                  T17PACKET:                              ;COMMAND PACKET FOR TEST
032640  100004                                  .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
032642  032650                                  .WORD   T17DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
032644  000000                                  .WORD   0
032646  000012                                  .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE

032650                                  T17DATA:                                ;CHARACTERISTICS DATA BLOCK
032650  032662                                  .WORD   T17BFR                  ;ADDRESS OF MESSAGE BUFFER
032652  000000                                  .WORD   0
032654  000024                                  .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
032656  000000                                  .WORD   0                       ;ESS,ENB,EAI,ERI
032660  000000                                  .WORD   0                       ;EXTENDED FEATURES UNIT NO. ETC.


                                        ;MESSAGE BUFFER FOR ALL TEST 6 COMMANDS

032662                                  T17BFR:                                 ;BEGIN MESSAGE BUFFER
```

```
 5460 032662  000000                    .WORD   0                       ;MESSAGE TYPE
 5461 032664  000000                    .WORD   0                       ;DATA FIELD LENGTH
 5462 032666  000000                    .WORD   0                       ;RBPCR
 5463 032670  000000                    .WORD   0                       ;XST0
 5464 032672  000000                    .WORD   0                       ;XST1
 5465 032674  000000                    .WORD   0                       ;XST2
 5466 032676  000000                    .WORD   0                       ;XST3
 5467 032700  000000                    .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
 5468 032702                    T17BFSTA: .BLKB 64.                       ;READ STATUS AND WRITE FIFO BUFFER
 5469 033002                    T17BEND:                                  ;END OF MESSAGE BUFFER
 5470                            ;
 5471                            ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
 5472                            ;
 5474 033002                            .BLKB   10-<.-TUV2A&7>
 5476 033010                    T17PK2:
 5477 033010  100006                    .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
 5478 033012  033020                    .WORD   T17DT2                  ;LOW ADDRESS OF DATA BLOCK
 5479 033014  000000                    .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
 5480 033016  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 5481
 5482 033020                    T17DT2:                                 ;DATA BLOCK
 5483 033020     000                    .BYTE   0                       ;BSEL0
 5484 033021     000                    .BYTE   0                       ;BSEL1
 5485 033022  000000                    .WORD   0                       ;SEL2
 5486 033024                            .BLKB   66.                     ;WRITE FIFO DATA OUTPUT BUFFER
 5487 033126  000000            T17DLY: .WORD   0                       ;HOLDS DELAY VALUE
 5488
 5489 033130                            ENDTST
      033130                                                    L10036:
      033130  104401                                                TRAP    C$ETST
 5490
```

```
                                        .SBTTL  TEST  2: INITIALIZE #2 TEST
                                ;+
                                ;
                                ;THIS TEST VERIFIES THAT WRITING INTO THE TSSR RETURNS THE
                                ;CONTROLLER TO ITS INITIALIZED STATE FROM VARIOUS CONDITIONS
                                ;
                                ;-
                                        BGNTST
                                                                                T2::
                                        CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                                        CLR     KTFLG           ;HOLD OFF KT11
                                        MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY ERROR MESSAGE

                                ;
                                ;
                                ;TEST 1
                                ;
                                ;
                                ;-

                                        JSR     PC,DSBINT               ;DISABLE INTERRUPTS
                                        MOV     #TST21ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
                                        JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
                                        MOV     #2,LOOPCNT              ;PERFORM 2 ITERATIONS
                                T21LOOP:
                                        JSR     PC,T21REST              ;SET COMMAND PACKET
                                        JSR     PC,T21RT2               ;SET UP OTHER COMMAND PACKET

                                ;*************************************************************************
                                ;
                                ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                ;
                                ;*************************************************************************

                                11$:    MOV     #65000.,T21DLY          ;SET DELAY ROUTINE
                                        JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
                                        BCS     20$                     ;BR IF INIT WAS OK
                                        DELAY   250                     ;DELAY FOR A REWIND TO FINISH
033216  012727  000250                                                          MOV     #250,(PC)+
033222  000000                                                                  .WORD   0
033224  013727  002116                                                          MOV     L$DLY,(PC)+
033230  000000                                                                  .WORD   0
033232  005367  177772                                                          DEC     -6(PC)
033236  001375                                                                  BNE     .-4
033240  005367  177756                                                          DEC     -22(PC)
033244  001367                                                                  BNE     .-20
                                        DEC     T21DLY                  ;BUMP COUNTER DOWN
                                        BNE     11$                     ;BR, IF MORE TIME TO GO
                                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                        MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
033262  104455                                                                  TRAP    C$ERDF
033264  000311                                                                  .WORD   201
033266  003550                                                                  .WORD   SFIERR
033270  011656                                                                  .WORD   SFIMSG
                                20$:
033272  012704  033440                    MOV     #T21PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
```

```
5543
5544                                    ;**********************************************************************
5545                                    ;
5546                                    ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
5547                                    ;
5548                                    ;**********************************************************************
5549
5550 033276  004737  010322                     JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
5551 033302  103407                             BCS     23$                     ;BR, IF COMMAND ISSUED OK
5552 033304  004737  020064                     JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5556 033310  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5557 033312                                     ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     033312  104456                                                                                     TRAP    C$ERHRD
     033314  000312                                                                                     .WORD   202
     033316  004754                                                                                     .WORD   WRTMSG
     033320  011656                                                                                     .WORD   SFIMSG
5558 033322                                 23$:
5559 033322  012765  000000  000000                 MOV     #0,TSSR(R5)             ;ISSUE A SOFT INITIALIZE
5560 033330  004737  017104                         JSR     PC,WAITF                ;WAIT FOR JUST THE SSR BIT TO SET
5561 033334  016501  000000                         MOV     TSSR(R5),R1             ;READ THE TSSR BACK
5562 033340  010102                                 MOV     R1,R2                   ;WORK REGISTER
5563 033342  042702  176377                         BIC     #^C<HIADDR>,R2          ;CLEAR OUT OTHER BITS
5564 033346  052702  002200                         BIS     #SSR!NBA,R2             ;SOME OF THE BITS THAT SHOULD BE SET
5565 033352  032701  000100                         BIT     #OFL,R1                 ;IS OFF LINE BIT SET
5566 033356  001012                                 BNE     38$                     ;BR, IF DRIVE IS OFF LINE
5567 033360  020102                         35$:    CMP     R1,R2                   ;EXPECTED (R2) = RECEIVED (R1)
5568 033362  001406                                 BEQ     37$                     ;BR, IF THEY ARE EQUAL (OK)
5569 033364  004737  020064                         JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5573 033370                                         ERRHRD  ERRNO,T21AM3,EXPREC     ;"ERROR TRYING TO INIT AFTER WRITE MISC.
     033370  104456                                                                                     TRAP    C$ERHRD
     033372  000313                                                                                     .WORD   203
     033374  033665                                                                                     .WORD   T21AM3
     033376  016330                                                                                     .WORD   EXPREC
5574 033400                                 37$:    CKLOOP                          ;LOOP IF SELECTED
     033400  104406                                                                                     TRAP    C$CLP1
5575 033402  000406                                 BR      40$                     ;SKIP OVER OFF-LINE STUFF
5576 033404                                 38$:
5580 033404                                         ERRDF   ERRNO,T21OFL,EXPREC     ;DRIVE IS OFF LINE
     033404  104455                                                                                     TRAP    C$ERDF
     033406  000314                                                                                     .WORD   204
     033410  033765                                                                                     .WORD   T21OFL
     033412  016330                                                                                     .WORD   EXPREC
5581 033414  004737  020136                         JSR     PC,CKDROP               ;TRY AND DROP UNIT
5582 033420  004737  017340                 40$:    JSR     PC,TSTLOOP              ;DO WE NEED TO ITERATE TEST
5583 033424  103002                                 BCC     63$                     ;BR, IF NO LOOP REQUIRED
5584 033426  000137  033172                         JMP     T21LOOP                 ;EXECUTE AGAIN
5585 033432                                 63$:    EXIT    TST                     ;ALL DONE THIS TEST
     033432  104432                                                                                     TRAP    C$EXIT
     033434  000526                                                                                     .WORD   L10045-.
```

```
 5587                                   ;+
 5588                                   ;LOCAL STORAGE FOR THIS TEST
 5589                                   ;-
 5591 033436                                    .BLKB   10-<.-TUV2A&7>
 5593 033440                            T21PACKET:                      ;COMMAND PACKET FOR TEST
 5594 033440  100004                            .WORD   100004          ;WRITE CHARACTERISTICS COMMAND, WITH, ACK
 5595 033442  033450                            .WORD   T21DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
 5596 033444  000000                            .WORD   0
 5597 033446  000012                            .WORD   10.             ;STARTING VALUE OF BLOCK SIZE
 5598 033450                            T21DATA:                        ;CHARACTERISTICS DATA BLOCK
 5599 033450  033460                            .WORD   T21BFR          ;ADDRESS OF MESSAGE BUFFER
 5600 033452  000000                            .WORD   0
 5601 033454  000024                            .WORD   20.             ;LENGTH OF MESSAGE BUFFER
 5602 033456  000000                            .WORD   0
 5603 033460                            T21BFR: .BLKW   25.             ;MESSAGE BUFFER
 5604                                   ;
 5605                                   ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
 5606                                   ;
 5608 033542                                    .BLKB   10-<.-TUV2A&7>
 5610 033550                            T21PK2:
 5611 033550  100206                            .WORD   100206          ;WRITE SUB SYS MEM COMMAND, IE AND ACK
 5612 033552  033560                            .WORD   T21BF2          ;ADDRESS OF SELECT BLOCK DATA
 5613 033554  000000                            .WORD   0
 5614 033556  000006                            .WORD   6.              ;SIZE OF DATA PACKET
 5615
 5616                                           .EVEN
 5617 033560                            T21BF2:
 5618 033560      000                   T21BS0: .BYTE   0               ;BSEL0 AREA --- "COMMAND" BYTE
 5619 033561      000                   T21BS1: .BYTE   0               ;BSEL1 AREA
 5620 033562  000000                    T21S2:  .WORD   0               ;SEL 2 AREA
 5621 033564  000000                    T21S3:  .WORD   0               ;DATA AREA
 5622 033566  000000                    T21DLY: .WORD   0               ;DELAY COUNTER
 5623
 5624
 5625
```

```
                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

033570  127  122  111  T21SSR: .ASCIZ  'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
033665  124  123  123  T21AM3: .ASCIZ  'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
033765  104  162  151  T21OFL: .ASCIZ  'Drive is OFFLINE'
034006  111  156  151  TST21ID:        .ASCIZ  'Initialization #2'
                              .EVEN
                                        ;+
                                        ;
                                        ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
                                        ;WRITE SUBSYSTEM MEMORY COMMAND
                                        ;
                                        ;-

034030                                  T21REST:
034030                                        SAVREG                          ;SAVE THE REGISTERS
034034  012701  033440                      MOV     #T21PACKET,R1             ;START OF THE PACKET
034040  012721  100004                      MOV     #100004,(R1)+             ;WRITE SUBSYSTEM MEM. WITH ACK,
034044  012721  033450                      MOV     #T21DATA,(R1)+            ;ADDRESS OF CHARAISTICS DATA BLOCK
034050  005021                            CLR     (R1)+                     ;EXTENDED ADDRESS
034052  012721  000010                      MOV     #8.,(R1)+                 ;SIZE OF DATA BLOCK IN BYTES
034056  012721  033460                      MOV     #T21BFR,(R1)+             ;ADDRESS OF MESSAGE BUFFER
034062  005021                            CLR     (R1)+
034064  012721  000024                      MOV     #20.,(R1)+                ;LENGTH OF MESSAGE BUFFER
034070  005021                            CLR     (R1)+
034072  005011                            CLR     (R1)
034074  012702  000020                      MOV     #20,R2                    ;NUMBER OF LOCATIONS TO BE CLEARED
034100  012762  177777  033460  64$:       MOV     #177777,T21BFR(R2)        ;ALL ONES TO MESSAGE BUFFER
034106  005742                            TST     -(R2)                     ;NEXT LOCATION
034110  020227  000000                      CMP     R2,#0                     ;CHECK R2 FOR ZERO
034114  001371                            BNE     64$                       ;BR, IF NOT AT ZERO YET
034116  000207                            RTS     PC                        ;RETURN
034120                                  T21RT2:
034120                                        SAVREG                          ;SAVE THE REGISTERS
034124  012701  033550                      MOV     #T21PK2,R1                ;START OF THE PACKET
034130  012721  100206                      MOV     #100206,(R1)+             ;WRITE SUBSYSTEM MEM. WITH ACK, IE
034134  012721  033560                      MOV     #T21BF2,(R1)+             ;ADDRESS OF DATA BLOCK
034140  005021                            CLR     (R1)+                     ;EXTENDED ADDRESS
034142  012721  000006                      MOV     #6.,(R1)+                 ;SIZE OF DATA BLOCK IN BYTES
034146  005021                            CLR     (R1)+
034150  012701  033560                      MOV     #T21BF2,R1                ;ADDRESS OF DATA FOR WRT SUB SYS MEM
034154  005021                            CLR     (R1)+
034156  005011                            CLR     (R1)
034160  000207                            RTS     PC                        ;RETURN
034162                                        ENDTST
034162                                                              L10045:
034162  104401                                                              TRAP    C$ETST
```

```
                                        .SBTTL  TEST  3: OFF-LINE AND REJECT REWIND
                                ;+
                                ;
                                ;THIS TEST VERIFIES BASIC TAPE-MOTION COMMAND DECODING AND BASIC
                                ;OPERATION OF THE REWIND POSITIONING COMMAND. IT DOES NOT
                                ;NECESSARILY DEMONSTRATE THAT THE TRANSPORT CAN BE REWOUND FROM AN
                                ;ARBITRARY POSITION ON THE TAPE. SUBSEQUENT TESTS IMPLICITLY
                                ;CHECK THE OPERATION OF THE REWIND COMMAND SINCE THEY MUST
                                ;TYPICALLY REWIND THE TAPE IN THE NORMAL COURSE OF THEIR TEST
                                ;SEQUENCES. THE TEST CONSISTS OF THE FOLLOWING ONE SUBTEST
                                ;
                                ;
                                ;-
034164                                  BGNTST
034164                                                                  T3::
034164  005037  002170                    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
034170  005037  003100                    CLR     KTFLG           ;HOLD OFF KT11
034174  012737  005755  002146            MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY ERROR MESSAGE
034202  004737  017040                    JSR     PC,DSBINT               ;DISABLE INTERRUPTS
034206  012700  035322                    MOV     #TST22ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
034212  004737  017372                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
034216  012737  000002  002164            MOV     #2,LOOPCNT              ;PERFORM 2 ITERATIONS
                                ;+
                                ;
```

```
                                T22LOOP:
                                ;+
                                ;
                                ;TEST 3, SUBTEST 1
                                ;
                                ;VERIFIES THAT A REWIND COMMAND WITH CVC=1 CLEARS VCK
                                ;AND RETURNS PROPER STATUS IN THE MESSAGE BUFFER.
                                ;
                                ;
                                ;-
                                        BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
                                                                                        T3.1:
                                                                                TRAP    C$BSUB
                                        JSR     PC,T22REST              ;SET COMMAND PACKET
                                        JSR     PC,T22RT2               ;SET UP OTHER COMMAND PACKET

                                ;*************************************************************************
                                ;
                                ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                ;
                                ;*************************************************************************

                                        JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
                                        BCS     20$                     ;BR IF INIT WAS OK
                                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                        MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
                                                                                TRAP    C$ERDF
                                                                                .WORD   301
                                                                                .WORD   SFIERR
                                                                                .WORD   SFIMSG
                                20$:
                                        MOV     #T22PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS

                                ;*************************************************************************
                                ;
                                ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                ;
                                ;*************************************************************************

                                        JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
                                        BCS     65$                     ;BR, IF COMMAND ISSUED OK
                                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                                                                                TRAP    C$ERHRD
                                                                                .WORD   302
                                                                                .WORD   WRTMSG
                                                                                .WORD   SFIMSG
                                65$:    MOV     #142010,T22PK2          ;POSITION COMMAND (REWIND MODE) CVC=1
                                        MOV     #T22PK2,R4              ;R4 = POINTER TO PACKET
                                        MOV     R4,TSDB(R5)             ;ISSUE COMMAND
                                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                                        MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
                                        MOV     #SSR,R2                 ;SET UP EXPECTED
                                        CMP     R1,R2                   ;ARE THEY EQUAL
                                        BEQ     80$                     ;BR, IF OK ESP. FUNCTION REJECT
```

```
5757 034350  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5761 034354                                ERRHRD  ERRNO,T22RWJ,EXPREC     ;TSSR INCORRECT AFTER TAPE MOTION CMD
     034354  104456                                                                TRAP    C$ERHRD
     034356  000457                                                                .WORD   303
     034360  035174                                                                .WORD   T22RWJ
     034362  016330                                                                .WORD   EXPREC
5762 034364                         80$:    CKLOOP                          ;LOOP IF SELECTED
     034364  104406                                                                TRAP    C$CLP1
5763 034366  012703  034512                MOV     #T22BFR,R3              ;POINTER TO MESSAGE BUFFER
5764 034372  016301  000006                MOV     XSTO(R3),R1             ;PICK UP XSTO FROM MESSAGE BUFFER
5765 034376  010102                        MOV     R1,R2                   ;SET UP EXPECTED
5766 034400  042702  000020                BIC     #BIT4,R2                ;VCK SHOULD BE CLEAR
5767 034404  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
5768 034406  001406                        BEQ     90$                     ;BR, IF OK (GOOD)
5769 034410  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5773 034414                                ERRHRD  ERRNO,T22VCK,EXPREC     ;VCK WASN'T CLEAR (BAD)
     034414  104456                                                                TRAP    C$ERHRD
     034416  000460                                                                .WORD   304
     034420  035247                                                                .WORD   T22VCK
     034422  016330                                                                .WORD   EXPREC
5774 034424                         90$:
5775 034424                                ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
     034424                                                                     L10047:
     034424  104403                                                                TRAP    C$ESUB
5776 034426  023727  002170  000031                CMP     FATFLG,#25.             ;IS ERROR COUNT AT 25
5777 034434  002402                                BLT     999$                    ;BR, IF LESS THAN 25
5778 034436  004737  020136                        JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
5779 034442                         999$:
5780 034442  004737  017340                JSR     PC,TSTLOOP              ;DO WE NEED TO ITERATE TEST
5781 034446  103002                        BCC     163$                    ;BR, IF NO LOOP REQUIRED
5782 034450  000137  034224                JMP     T22LOOP                 ;EXECUTE AGAIN
5783 034454                         163$:   EXIT    TST                     ;ALL DONE THIS TEST
     034454  104432                                                                TRAP    C$EXIT
     034456  001036                                                                .WORD   L10046-.
```

```
                                ;+
                                ;LOCAL STORAGE FOR THIS TEST
                                ;-
034460                                  .BLKB   10-<.-TUV2A87>
034470                          T22PACKET:                      ;COMMAND PACKET FOR TEST
034470  100204                          .WORD   100204          ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
034472  034500                          .WORD   T22DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
034474  000000                          .WORD   0
034476  000012                          .WORD   10.             ;STARTING VALUE OF BLOCK SIZE
034500                          T22DATA:                        ;CHARACTERISTICS DATA BLOCK
034500  034512                          .WORD   T22BFR          ;ADDRESS OF MESSAGE BUFFER
034502  000000                          .WORD   0
034504  000024                          .WORD   20.             ;LENGTH OF MESSAGE BUFFER
034506  000000                          .WORD   0
034510  000007                          .WORD   7               ;SELECT DRIVE 7
034512                          T22BFR: .BLKW   25.             ;MESSAGE BUFFER
                                ;
                                ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
                                ;
034574                                  .BLKB   10-<.-TUV2A87>
034600                          T22PK2:
034600  100206                          .WORD   100206          ;WRITE SUB SYS MEM COMMAND, IE AND ACK
034602  034610                          .WORD   T22BF2          ;ADDRESS OF SELECT BLOCK DATA
034604  000000                          .WORD   0
034606  000006                          .WORD   6.              ;SIZE OF DATA PACKET

                                        .EVEN
034610                          T22BF2:
034610     000                  T22BS0: .BYTE   0               ;BSEL0 AREA
034611     000                  T22BS1: .BYTE   0               ;BSEL1 AREA
034612  000000                  T22S2:  .WORD   0               ;SEL 2 AREA
034614  000000                  T22S3:  .WORD   0               ;DATA AREA
                                ;
                                ;
                                        .EVEN
                                ;TAPE MOTION PACKET COMMAND VALUES
034616  100201                  T22RD:  .WORD   100201          ;READ TAPE FORWARD
034620  100205                  T22WRT: .WORD   100205          ;WRITE TAPE FORWARD
034622  100210                  T22POS: .WORD   100210          ;POSITION TAPE
034624  100211                  T22FOR: .WORD   100211          ;FORMAT TAPE
034626  177777                          .WORD   177777          ;END OF DATA
```

```
 5833
 5834                                   ;+
 5835                                   ;LOCAL TEXT MESSAGES FOR TEST
 5836                                   ;-
 5837
 5838 034630   127   122   111  T22SSR: .ASCIZ  'WRITE MISCELLANEOUS CONTROL/READ STATUS Command Not Accepted'
 5839 034725   124   123   123  T22AM3: .ASCIZ  'TSSR Init. Failed After WRITE MISCELLANEOUS CONRTOL/READ STATUS'
 5840 035025   104   162   151  T22OFL: .ASCIZ  'Drive 7 Select Failed To Set "OFL" In TSSR'
 5841 035100   124   123   123  T22TM:  .ASCIZ  'TSSR Incorrect After Tape Motion Command To Off-Line Device'
 5842 035174   124   123   123  T22RWJ: .ASCIZ  'TSSR Not Correct After REWIND With VCK Set'
 5843 035247   103   126   103  T22VCK: .ASCIZ  'CVC Set, Didn''t Reset VCK In Message Buffer'
 5844 035322   117   146   146  TST22ID:        .ASCIZ  'Off-Line And Reject Rewind'
 5845                                   .EVEN
 5846                                   ;+
 5847                                   ;
 5848                                   ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
 5849                                   ;WRITE SUBSYSTEM MEMORY COMMAND
 5850                                   ;
 5851                                   ;-
 5852
 5853 035356                            T22REST:
 5854 035356                                    SAVREG                          ;SAVE THE REGISTERS
 5855 035362  012701  034470                      MOV     #T22PACKET,R1         ;START OF THE PACKET
 5856 035366  012721  100204                      MOV     #100204,(R1)+         ;WRITE SUBSYSTEM MEM. WITH ACK, IE
 5857 035372  012721  034500                      MOV     #T22DATA,(R1)+        ;ADDRESS OF CHARAISTICS DATA BLOCK
 5858 035376  005021                              CLR     (R1)+                 ;EXTENDED ADDRESS
 5859 035400  012721  000012                      MOV     #10.,(R1)+            ;SIZE OF DATA BLOCK IN BYTES
 5860 035404  012721  034512                      MOV     #T22BFR,(R1)+         ;ADDRESS OF MESSAGE BUFFER
 5861 035410  005021                              CLR     (R1)+
 5862 035412  012721  000024                      MOV     #20.,(R1)+            ;LENGTH OF MESSAGE BUFFER
 5863 035416  005021                              CLR     (R1)+
 5864 035420  012711  000007                      MOV     #7,(R1)               ;SELECT DRIVE SEVEN
 5865 035424  012702  000020                      MOV     #20,R2                ;NUMBER OF LOCATIONS TO BE CLEARED
 5866 035430  012762  177777  034512  64$:       MOV     #177777,T22BFR(R2)    ;ALL ONES TO MESSAGE BUFFER
 5867 035436  005742                              TST     -(R2)                 ;BUMP R2 DOWN
 5868 035440  020227  000000                      CMP     R2,#0                 ;IS R2 AT ZERO YET
 5869 035444  001371                              BNE     64$                   ;KEEP GOING UNTIL DONE
 5870 035446  000207                              RTS     PC                    ;RETURN
 5871 035450                            T22RT2:
 5872 035450                                    SAVREG                          ;SAVE THE REGISTERS
 5873 035454  012701  034600                      MOV     #T22Pk2,R1            ;START OF THE PACKET
 5874 035460  012721  100206                      MOV     #100206,(R1)+         ;WRITE SUBSYSTEM MEM. WITH ACK, IE
 5875 035464  012721  034610                      MOV     #T22BF2,(R1)+         ;ADDRESS OF DATA BLOCK
 5876 035470  005021                              CLR     (R1)+                 ;EXTENDED ADDRESS
 5877 035472  012721  000006                      MOV     #6.,(R1)+             ;SIZE OF DATA BLOCK IN BYTES
 5878 035476  005021                              CLR     (R1)+
 5879 035500  012701  034610                      MOV     #T22BF2,R1            ;POINT TO DATA SEL AREA
 5880 035504  005021                              CLR     (R1)+
 5881 035506  005011                              CLR     (R1)
 5882 035510  005011                              CLR     (R1)                  ;LAST LOC TO BE CLEARED
 5883 035512  000207                              RTS     PC                    ;RETURN
 5884 035514                                    ENDTST
      035514                                                          L10046:
      035514  104401                                                           TRAP    CSETST
```

```
                                        .SBTTL  TEST  4: BASIC WRITE DATA
                                ;+
                                ;
                                ;THIS TEST VERIFIES THAT THE WRITE DATA (NEXT) COMMAND OPERATES
                                ;PROPERLY, UP TO THE POINT OF CHECKING THAT THE DATA WAS ACTUALLY
                                ;WRITTEN ONTO THE TAPE CORRECTLY. CHECKING IN THIS TEST IS
                                ;LIMITED TO VERIFYING THAT THE COMMAND TERMINATED CORRECTLY WITH
                                ;THE CORRECT REGISTER, MESSAGE BUFFER AND RAM CONTENTS.
                                ;
                                ;THE TEST CONSISTS OF THE FOLLOWING 5 SUBTESTS
                                ;
                                ;
                                ;
                                ;-
035516                                  BGNTST
035516                                                                  T4::
035516  005037  002170                    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
035522  005037  003100                    CLR     KTFLG           ;HOLD OFF KT11
035526  012737  005755  002146            MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY ERROR MESSAGE
035534  005037  003102                    CLR     KTENABLE                        ;TURN OFF KT11
035540  004737  020230                    JSR     PC,KTOFF                        ;TURN OFF KT11
035544  004737  017040                    JSR     PC,DSBINT               ;DISABLE INTERRUPTS
035550  012700  041251                    MOV     #TST23ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
035554  004737  017372                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
035560  012737  000001  002164            MOV     #1,LOOPCNT              ;PERFORM 1 ITERATIONS
                                ;+
                                ;
```

```
 5917 035566                            T23LOOP:
 5918                                   ;+
 5919                                   ;
 5920                                   ;TEST 4, SUBTEST 1
 5921                                   ;
 5922                                   ;VERIFIES THAT WRITE DATA COMMANDS WITH CVC=1 AND THE
 5923                                   ;SWAP BYTES (SWB) BIT CLEAR OPERATES PROPERLY. THE
 5924                                   ;BYTE COUNT (RECORD SIZE) VARIES FROM 20 THROUGH 64K
 5925                                   ;IN VARYING INCREMENTS (DEPENDING UPON WHETHER OR NOT
 5926                                   ;THE DIAGNOSTIC IS RUNNING IN THE LONG VERIFICATION
 5927                                   ;MODE). THE TAPE IS NOT REWOUND BETWEEN SUCCESSIVE
 5928                                   ;RECORDS BUT IS REWOUND AFTER THE FINAL RECORD IS
 5929                                   ;WRITTEN. AN INCREMENTING COUNT PATTERN IS SUPPLIED
 5930                                   ;IN THE DATA BUFFER. AFTER EACH BLOCK IS WRITTEN, THE
 5931                                   ;TSSR AND TSBA REGISTERS, AND THE MESSAGE BUFFER.
 5932                                   ;
 5933                                   ;
 5934                                   ;
 5935                                   ;-
 5936 035566                                    BGNSUB                  ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
      035566                                                                    T4.1:
      035566  104402                                                                TRAP    C$BSUB
 5937 035570  013701  003074                    MOV     FRESIZ,R1       ;GET SIZE OF AVAILIABLE MEMORY
 5938 035574  000241                            CLC                     ;CLEAR THE CARRY BIT
 5939 035576  006101                            ROL     R1              ;MAKE INTO BYTES
 5940 035600  010137  040010                    MOV     R1,T23RSZ       ;STORE IN TEST FOR USE LATER
 5941 035604  004737  041266                    JSR     PC,T23REST      ;SET COMMAND PACKET
 5942 035610  004737  041422                    JSR     PC,T23RT3       ;RESTORE PACKET
 5943 035614  004737  041360                    JSR     PC,T23RT2       ;SET UP OTHER COMMAND PACKET
 5944
 5945                                   ;*****************************************************************
 5946                                   ;
 5947                                   ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
 5948                                   ;
 5949                                   ;*****************************************************************
 5950
 5951 035620  004737  016630                    JSR     PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
 5952 035624  103407                            BCS     20$             ;BR IF INIT WAS OK
 5953 035626  004737  020064                    JSR     PC,FATCHK       ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5957 035632  010001                            MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
 5958 035634                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
      035634  104455                                                                TRAP    C$ERDF
      035636  000621                                                                .WORD   401
      035640  003550                                                                .WORD   SFIERR
      035642  011656                                                                .WORD   SFIMSG
 5959 035644                            20$:
 5960 035644  012704  037650                    MOV     #T23PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
 5961
 5962                                   ;*****************************************************************
 5963                                   ;
 5964                                   ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
 5965                                   ;
 5966                                   ;*****************************************************************
 5967
 5968 035650  004737  010322                    JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
 5969 035654  103407                            BCS     23$             ;BR, IF COMMAND ISSUED OK
 5970 035656  004737  020064                    JSR     PC,FATCHK       ;INC AND CHECK FOR MORE THAN 25 ERRORS
```

```
 5974 035662  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5975 035664                                 ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      035664  104456                                                                TRAP    C$ERHRD
      035666  000622                                                                .WORD   402
      035670  004754                                                                .WORD   WRTMSG
      035672  011656                                                                .WORD   SFIMSG
 5976 035674                        23$:    CKLOOP                          ;LOOP IF SELECTED
      035674  104406                                                                TRAP    C$CLP1
 5977
 5978                               ;*************************************************************************
 5979                               ;
 5980                               ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
 5981                               ;
 5982                               ;*************************************************************************
 5983
 5984 035676  004737  010424                JSR     PC,REWIND               ;CALL THE TAPE REWIND
 5985 035702  012703  000024                MOV     #20.,R3                 ;STARTING RECORD SIZE
 5986 035706  013737  003072  040002  65$:  MOV     FREE,T23WB              ;STARTING WRITE BUFFER ADDRESS
 5987
 5988                               ;*************************************************************************
 5989                               ;
 5990                               ;WRITE DATA,CVC=1,ACK COMMAND
 5991                               ;
 5992                               ;*************************************************************************
 5993
 5994 035714  012737  140005  040000        MOV     #140005,T23PK3          ;WRITE DATA,CVC=1,ACK COMMAND
 5995 035722  012704  040000                MOV     #T23PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
 5996 035726  010300                        MOV     R3,R0                   ;SET PATTERN IN CORRECT REGISTER
 5997 035730  004737  020356                JSR     PC,FILLMEM              ;FILL MEMORY WITH RECORD SIZE
 5998 035734  010337  040006                MOV     R3,T23SZ                ;SET UP RECORD SIZE IN PACKET
 5999 035740  010465  177776                MOV     R4,TSDB(R5)             ;ISSUE COMMAND
 6000 035744  004737  017104                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 6001 035750  016501  000000                MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
 6002 035754  012702  000200                MOV     #SSR,R2                 ;SET UP EXPECTED
 6003 035760  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
 6004 035762  001406                        BEQ     80$                     ;BR, IF OK
 6005 035764  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6010 035770                                ERRHRD  ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER WRITE DATA
      035770  104456                                                                TRAP    C$ERHRD
      035772  000623                                                                .WORD   403
      035774  005011                                                                .WORD   WRTERR
      035776  011670                                                                .WORD   PKTSSR
 6011 036000                        80$:    CKLOOP                          ;LOOP IF SELECTED
      036000  104406                                                                TRAP    C$CLP1
 6012 036002  016501  177776                MOV     TSBA(R5),R1             ;GET TSBA CONTENTS
 6013 036006  012702  040000                MOV     #T23PK3,R2              ;SET UP EXPECTED
 6014 036012  020102                85$:    CMP     R1,R2                   ;ARE THEY EQUAL
 6015 036014  001406                        BEQ     90$                     ;BR, IF TSBA IS CORRECT
 6016 036016  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6020 036022                                ERRHRD  ERRNO,T23BA,EXPREC      ;TSBA WAS NOT CORRECT AFTER WRITE DATA
      036022  104456                                                                TRAP    C$ERHRD
      036024  000624                                                                .WORD   404
      036026  041110                                                                .WORD   T23BA
      036030  016330                                                                .WORD   EXPREC
 6021 036032                        90$:    CKLOOP                          ;LOOP IF SELECTED
      036032  104406                                                                TRAP    C$CLP1
 6022 036034  062703  001750        115$:   ADD     #1000.,R3               ;NEXT RECORD SIZE/DATA PATTERN
```

```
 6023 036040  020337  040010                CMP     R3,T23RSZ               ;IS R3 OVER MAX RECORD SIZE
 6024 036044  002005                        BGE     120$                    ;IF RECORD SIZE IS TOO BIG QUIT
 6025 036046  020327  177776                CMP     R3,#65534.              ;END OF SUBTEST MAX RECORD SIZE
 6026 036052  001402                        BEQ     120$                    ;BR, IF COMPLETED
 6027 036054  000137  035706                JMP     65$                     ;DO MORE RECORDS
 6028 036060                         120$:
 6029 036060  004737  041422                JSR     PC,T23RT3               ;RESTORE PACKET
 6030 036064  004737  041360                JSR     PC,T23RT2               ;CLEAN UP PACKET
 6031 036070  004737  010424                JSR     PC,REWIND               ;ISSUE REWIND COMMAND WITH WAIT
 6032 036074  103407                        BCS     130$                    ;BR, IF TSSR IS OK (GOOD)
 6033 036076  010001                        MOV     R0,R1                   ;SAVE TSSR CONTENTS
 6034 036100  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6038 036104                                ERRHRD  ERRNO,T23RWN,PKTSSR     ;TSSR IS INCORRECT AFTER REWIND
      036104  104456                                                                TRAP    C$ERHRD
      036106  000625                                                                .WORD   405
      036110  040401                                                                .WORD   T23RWN
      036112  011670                                                                .WORD   PKTSSR
 6039 036114                         130$:
 6040 036114                                ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
      036114                                                                         L10051:
      036114  104403                                                                TRAP    C$ESUB
 6041 036116  023727  002170  000031              CMP     FATFLG,#25.               ;IS ERROR COUNT AT 25
 6042 036124  002402                              BLT     999$                      ;BR, IF LESS THAN 25
 6043 036126  004737  020136                      JSR     PC,CKDROP                 ;TRY TO DROP THE UNIT
 6044 036132                         999$:
```

```
                                        ;+
                                        ;
                                        ;TEST 4, SUBTEST 2
                                        ;
                                        ;VERIFIES THAT WRITE DATA COMMANDS WITH CVC=1 AND THE
                                        ;SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST
                                        ;SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2.
                                        ;THE RESULTS, EXCEPT FOR RAM CONTENTS, SHOULD BE THE SAME.
                                        ;
                                        ;
                                        ;
                                        ;-
036132                                          BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
036132                                                                                  T4.2:
036132  104402                                                                          TRAP    C$BSUB
036134  004737  041266                        JSR     PC,T23REST              ;SET COMMAND PACKET
036140  004737  041360                        JSR     PC,T23RT2               ;SET UP OTHER COMMAND PACKET

                                        ;*************************************************************************
                                        ;
                                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                        ;
                                        ;*************************************************************************

036144  004737  016630                        JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
036150  103407                                BCS     20$                     ;BR IF INIT WAS OK
036152  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
036156  010001                                MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
036160                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
036160  104455                                                                          TRAP    C$ERDF
036162  000626                                                                          .WORD   406
036164  003550                                                                          .WORD   SFIERR
036166  011656                                                                          .WORD   SFIMSG
036170                                  20$:
036170  012704  037650                        MOV     #T23PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS

                                        ;*************************************************************************
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                        ;
                                        ;*************************************************************************

036174  004737  010322                        JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
036200  103407                                BCS     23$                     ;BR, IF COMMAND ISSUED OK
036202  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
036206  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036210                                        ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
036210  104456                                                                          TRAP    C$ERHRD
036212  000627                                                                          .WORD   407
036214  004754                                                                          .WORD   WRTMSG
036216  011656                                                                          .WORD   SFIMSG
036220                                  23$:
036220  012703  000024                        MOV     #20.,R3                 ;STARTING RECORD SIZE
036224  013737  003072  040002          65$:  MOV     FREE,T23WB              ;STARTING WRITE BUFFER ADDRESS

                                        ;*************************************************************************
                                        ;
```

```
 6099                                   ;WRITE DATA,CVC=1,ACK,SWB COMMAND
 6100                                   ;
 6101                                   ;*************************************************************************
 6102
 6103 036232  012737  150005  040000          MOV     #150005,T23PK3          ;WRITE DATA,CVC=1,ACK,SWB COMMAND
 6104 036240  012704  040000                  MOV     #T23PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
 6105 036244  010300                          MOV     R3,R0                   ;SET PATTERN IN CORRECT REGISTER
 6106 036246  004737  020356                  JSR     PC,FILLMEM              ;FILL MEMORY WITH RECORD SIZE
 6107 036252  010337  040006                  MOV     R3,T23SZ                ;SET UP RECORD SIZE IN PACKET
 6108 036256  010465  177776                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
 6109 036262  004737  017104                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 6110 036266  016501  000000                  MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
 6111 036272  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
 6112 036276  020102                          CMP     R1,R2                   ;ARE THEY EQUAL
 6113 036300  001406                          BEQ     80$                     ;BR, IF OK
 6114 036302  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6118 036306                                  ERRHRD  ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER WRITE DATA
      036306  104456                                                                  TRAP    C$ERHRD
      036310  000630                                                                  .WORD   408
      036312  005011                                                                  .WORD   WRTERR
      036314  011670                                                                  .WORD   PKTSSR
 6119 036316                          80$:    CKLOOP                          ;LOOP IF SELECTED
      036316  104406                                                                  TRAP    C$CLP1
 6120 036320  016501  177776                  MOV     TSBA(R5),R1             ;GET TSBA CONTENTS
 6121 036324  012702  040000                  MOV     #T23PK3,R2              ;SET UP EXPECTED
 6122 036330  020102                  85$:    CMP     R1,R2                   ;ARE THEY EQUAL
 6123 036332  001406                          BEQ     90$                     ;BR, IF TSBA IS CORRECT
 6124 036334  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6128 036340                                  ERRHRD  ERRNO,T23BA,EXPREC      ;TSBA WAS NOT CORRECT AFTER WRITE DATA
      036340  104456                                                                  TRAP    C$ERHRD
      036342  000631                                                                  .WORD   409
      036344  041110                                                                  .WORD   T23BA
      036346  016330                                                                  .WORD   EXPREC
 6129 036350                          90$:    CKLOOP                          ;LOOP IF SELECTED
      036350  104406                                                                  TRAP    C$CLP1
 6130 036352  020327  007376                  CMP     R3,#7376                ;ONLY CHECK RAM UNTIL ITS FULL
 6131 036356  002057                          BGE     115$                    ;IT WRAPS AROUND ETC.
 6132 036360  004737  041360                  JSR     PC,T23RT2               ;MAKE SURE PACKET AND DATA ARE CLEAN
 6133 036364  012737  000400  040014          MOV     #256.,T23S2             ;STARTING RAM ADDRESS
 6134 036372  112737  000000  040012          MOVB    #0,T23BS0               ;STOP INTERNAL TUV05 DIAGNOSTICS
 6135 036400  112737  000000  040013          MOVB    #0,T23BS1               ;SIZE OF RAM READ
 6136 036406  012704  037760                  MOV     #T23PK2,R4              ;SET R4 WITH PACKET ADDRESS
 6137 036412  010465  177776                  MOV     R4,TSDB(R5)             ;ISSUE WRITE SUB SYS MEM COMMAND
 6138 036416  004737  017220                  JSR     PC,CHKTSSR              ;CHECK TSSR AND WAIT FOR SSR TO SET
 6139 036422  103407                          BCS     92$                     ;BR, IF NO ERRORS IN TSSR
 6140 036424  010001                          MOV     R0,R1                   ;SAVE TSSR
 6141 036426  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6145 036432                                  ERRHRD  ERRNO,T23WSS,PKTSSR     ;TSSR BAD AFTER WRITE SUB SYS MEM
      036432  104456                                                                  TRAP    C$ERHRD
      036434  000632                                                                  .WORD   410
      036436  041162                                                                  .WORD   T23WSS
      036440  011670                                                                  .WORD   PKTSSR
 6146 036442                          92$:    CKLOOP                          ;LOOP IF SELECTED
      036442  104406                                                                  TRAP    C$CLP1
 6147 036444  004737  041360                  JSR     PC,T23RT2               ;MAKE SURE PACKET AND DATA ARE CLEAN
 6148 036450  012737  000400  040014          MOV     #256.,T23S2             ;STARTING RAM ADDRESS
 6149 036456  112737  000001  040012          MOVB    #1,T23BS0               ;READ RAM COMMAND FOR WRITE SUB SYS M.
```

```
 6150 036464  112737  000002  040013          MOVB    #2,T23BS1               ;SIZE OF RAM READ
 6151 036472  012704  037760                  MOV     #T23PK2,R4              ;SET R4 WITH PACKET ADDRESS
 6152 036476  010465  177776          95$:    MOV     R4,TSDB(R5)             ;ISSUE WRITE SUB SYS MEM CMD (READ RAM)
 6153 036502  004737  017220                  JSR     PC,CHKTSSR              ;CHECK TSSR AND WAIT FOR SSR TO SET
 6154 036506  103403                          BCS     115$                    ;BR, IF NO ERRORS IN TSSR
 6155 036510  010001                          MOV     R0,R1                   ;SAVE TSSR
 6156 036512  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6160 036516  062703  001750          115$:   ADD     #1000.,R3               ;NEXT RECORD SIZE/DATA PATTERN
 6161 036522  020337  040010                  CMP     R3,T23RSZ               ;IS R3 OVER MAX RECORD SIZE
 6162 036526  002005                          BGE     120$                    ;IF RECORD SIZE IS TOO BIG QUIT
 6163 036530  020327  177776                  CMP     R3,#65534.              ;END OF SUBTEST MAX RECORD SIZE
 6164 036534  001402                          BEQ     120$                    ;BR, IF COMPLETED
 6165 036536  000137  036224                  JMP     65$                     ;DO MORE RECORDS
 6166 036542                          120$:
 6167 036542  004737  041360                  JSR     PC,T23RT2               ;CLEAN UP PACKET
 6168 036546  004737  010424                  JSR     PC,REWIND               ;ISSUE REWIND COMMAND WITH WAIT
 6169 036552  103407                          BCS     130$                    ;BR, IF TSSR IS OK (GOOD)
 6170 036554  010001                          MOV     R0,R1                   ;SAVE TSSR CONTENTS
 6171 036556  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6175 036562                                  ERRHRD  ERRNO,T23RWN,PKTSSR     ;TSSR IS INCORRECT AFTER REWIND
      036562  104456                                                                  TRAP    C$ERHRD
      036564  000634                                                                  .WORD   412
      036566  040401                                                                  .WORD   T23RWN
      036570  011670                                                                  .WORD   PKTSSR
 6176 036572                          130$:
 6177 036572                                  ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
      036572                                                                       L10052:
      036572  104403                                                                TRAP    C$ESUB
 6178 036574  023727  002170  000031          CMP     FATFLG,#25.             ;IS ERROR COUNT AT 25
 6179 036602  002402                          BLT     999$                    ;BR, IF LESS THAN 25
 6180 036604  004737  020136                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 6181 036610                          999$:
```

```
                                ;+
                                ;
                                ;TEST 4, SUBTEST 3
                                ;
                                ;VERIFIES THAT A WRITE COMMAND WITH AN ILLEGAL MODE
                                ;FIELD OR AN ILLEGAL BUFFER ADDRESS IS REJECTED WITH
                                ;THE PROPER ERROR STATUS AND THAT TAPE DOES NOT MOVE
                                ;
                                ;
                                ;-
                                        BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
                                                                                        T4.3:
036610  104402                                                                          TRAP    C$BSUB
036612  004737  041266                  JSR     PC,T23REST              ;SET COMMAND PACKET
036616  004737  041422                  JSR     PC,T23RT3               ;RESTORE PACKET
036622  004737  041360                  JSR     PC,T23RT2               ;SET UP OTHER COMMAND PACKET

                                ;*************************************************************************
                                ;
                                ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                ;
                                ;*************************************************************************

036626  004737  016630                  JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
036632  103407                          BCS     20$                     ;BR IF INIT WAS OK
036634  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
036640  010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
036642                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
036642  104455                                                                          TRAP    C$ERDF
036644  000635                                                                          .WORD   413
036646  003550                                                                          .WORD   SFIERR
036650  011656                                                                          .WORD   SFIMSG
036652                          20$:
036652  012704  037650                  MOV     #T23PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS

                                ;*************************************************************************
                                ;
                                ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                ;
                                ;*************************************************************************

036656  004737  010322                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
036662  103407                          BCS     23$                     ;BR, IF COMMAND ISSUED OK
036664  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
036670  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036672                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
036672  104456                                                                          TRAP    C$ERHRD
036674  000636                                                                          .WORD   414
036676  004754                                                                          .WORD   WRTMSG
036700  011656                                                                          .WORD   SFIMSG

                                ;*************************************************************************
                                ;
                                ;WRITE DATA, ACK, ILLEGAL BITS
                                ;
                                ;*************************************************************************
```

```
 6236
 6237 036702  012737  104405  040000  23$:    MOV     #104405,T23PK3          ;WRITE DATA, ACK, ILLEGAL BITS
 6238 036710  013737  003072  040002          MOV     FREE,T23WB              ;SET UP WRITE BUFFER ADDRESS
 6239 036716  062737  000001  040002          ADD     #1.T23WB                ;MAKE ADDRESS ODD (ILLEGAL)
 6240 036724  012737  000400  040006          MOV     #256.,T23SZ             ;SET UP BUFFER SIZE
 6241 036732  012704  040000                  MOV     #T23PK3,R4              ;R4 = POINTER TO PACKET
 6242 036736  010465  177776                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
 6243 036742  004737  017104                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 6244 036746  016501  000000                  MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
 6245 036752  012702  100206                  MOV     #SSR!SC!BIT1!BIT2,R2    ;SET UP EXPECTED
 6246 036756  020102                          CMP     R1,R2                   ;ARE THEY EQUAL
 6247 036760  001406                          BEQ     80$                     ;BR, IF OK ESP. FUNCTION REJECT
 6248 036762  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6252 036766                                  ERRHRD  ERRNO,T23TM,PKTSSR      ;TSSR INCORRECT AFTER WRITE COMMAND
      036766  104456                                                                  TRAP    C$ERHRD
      036770  000637                                                                  .WORD   415
      036772  040226                                                                  .WORD   T23TM
      036774  011670                                                                  .WORD   PKTSSR
 6253 036776                          80$:    CKLOOP                          ;LOOP IF SELECTED
      036776  104406                                                                  TRAP    C$CLP1
 6254 037000                                  ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
      037000                                                                       L10053:
      037000  104403                                                                  TRAP    C$ESUB
 6255 037002  023727  002170  000031          CMP     FATFLG,#25.                     ;IS ERROR COUNT AT 25
 6256 037010  002402                          BLT     999$                            ;BR, IF LESS THAN 25
 6257 037012  004737  020136                  JSR     PC,CKDROP                       ;TRY TO DROP THE UNIT
 6258 037016                          999$:
```

```
                                        ;+
                                        ;
                                        ;TEST 4, SUBTEST 4
                                        ;
                                        ;VERIFIES THAT A WRITE DATA COMMAND SPECIFYING A DATA
                                        ;BUFFER STARTING IN NONEXISTANT MEMORY TERMINATES WITH
                                        ;THE PROPER ERROR STATUS WITHOUT MOVING TAPE
                                        ;
                                        ;
                                        ;
                                        ;-

                                                BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
                                                                                                T4.4:
                                                                                                        TRAP    C$BSUB
                                                TST     KTFLG                   ;CHECK FOR KT-11
                                                BEQ     10$                     ;BR, IF NO KT-11
                                                JMP     90$                     ;SKIP THIS TEST IF KT-11
                                        10$:    JSR     PC,T23REST              ;SET COMMAND PACKET
                                                JSR     PC,T23RT3               ;RESTORE PACKET
                                                JSR     PC,T23RT2               ;SET UP OTHER COMMAND PACKET

                                        ;*************************************************************************
                                        ;
                                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                        ;
                                        ;*************************************************************************


                                                JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
                                                BCS     20$                     ;BR IF INIT WAS OK
                                                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                                MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
                                                ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
                                                                                                TRAP    C$ERDF
                                                                                                .WORD   416
                                                                                                .WORD   SFIERR
                                                                                                .WORD   SFIMSG
                                        20$:
                                                MOV     #T23PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS

                                        ;*************************************************************************
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                        ;
                                        ;*************************************************************************


                                                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
                                                BCS     23$                     ;BR, IF COMMAND ISSUED OK
                                                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                                                                                                TRAP    C$ERHRD
                                                                                                .WORD   417
                                                                                                .WORD   WRTMSG
                                                                                                .WORD   SFIMSG

                                        ;*************************************************************************
```

```
                                        ;
                                        ;WRITE DATA, ACK, CVC=1
                                        ;
                                        ;*************************************************************************

037122  005037  050216          23$:    CLR     T24DLY                  ;SET EXTENDED ADDRESS BITS TO 0
037126  012737  140005  040000  25$:    MOV     #140005,T23PK3          ;WRITE DATA, ACK, CVC=1
037134  012701  160000                  MOV     #160000,R1              ;START POSSIBLE NXM ADDRESS
037140  012702  177776                  MOV     #177776,R2              ;END POSSIBLE NXM ADDRESS
037144  004737  017260                  JSR     PC,XNXM                 ;CALL NXM FINDER ROUTINE
037150  103402                          BCS     76$                     ;BR IF NXM ADDRESS FOUND
037152  000137  037266                  JMP     90$                     ;JMP OVER CAN'T FIND NXM
037156  010137  040002          76$:    MOV     R1,T23WB                ;SET UP WRITE BUFFER ADDRESS
037162  013737  050216  040004        MOV     T24DLY,T23WB+2          ;HIGH ORDER ADDRESS BITS
037170  012737  000100  040006        MOV     #64.,T23SZ              ;SET UP BUFFER SIZE
037176  012704  040000                MOV     #T23PK3,R4              ;R4 = POINTER TO PACKET
037202  010465  177776                MOV     R4,TSDB(R5)             ;ISSUE COMMAND
037206  004737  017104                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
037212  016501  000000                MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
037216  012702  104210                MOV     #SC!NXM!SSR!BIT3,R2     ;SET UP EXPECTED
037222  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
037224  001417                        BEQ     80$                     ;BR, IF OK ESP. FUNCTION REJECT
037226  062737  000001  050216        ADD     #1,T24DLY               ;LOOK AT NEXT EXTENDED BITS WORTH OF MEM
037234  022737  000004  050216        CMP     #4,T24DLY               ;TOO MUCH MEMORY YET
037242  001402                        BEQ     168$                    ;BR, IF OVER 18 BIT ADDRESS
037244  000137  037126                JMP     25$                     ;TRY AGAIN (NEXT BUNCH OF MEMORY)
037250  004737  020064          168$:   JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
037254                                  ERRHRD  ERRNO,T23TM,PKTSSR      ;TSSR INCORRECT AFTER WRITE COMMAND
037254  104456                                                                  TRAP    C$ERHRD
037256  000642                                                                  .WORD   418
037260  040226                                                                  .WORD   T23TM
037262  011670                                                                  .WORD   PKTSSR
037264                          80$:    CKLOOP                          ;LOOP IF SELECTED
037264  104406                                                                  TRAP    C$CLP1
037266                          90$:
037266                                  ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
037266                                                                        L10054:
037266  104403                                                                  TRAP    C$ESUB
037270  023727  002170  000031          CMP     FATFLG,#25.                     ;IS ERROR COUNT AT 25
037276  002402                          BLT     999$                            ;BR, IF LESS THAN 25
037300  004737  020136                  JSR     PC,CKDROP                       ;TRY TO DROP THE UNIT
037304                          999$:
```

```
                                        ;+
                                        ;
                                        ;TEST 4, SUBTEST 5
                                        ;
                                        ;VERIFIES THAT A WRITE DATA COMMAND SPECIFYING A DATA
                                        ;BUFFER STARTING IN EXISTANT MEMORY BUT RUNNING INTO
                                        ;NONEXISTENT MEMORY TERMINATES WITH THE PROPER ERROR
                                        ;STATUS. A LARGE ENOUGH RECORD SIZE IS SPECIFIED SUCH
                                        ;THAT TAPE IS ACTUALLY MOVED AND WRITTEN.
                                        ;
                                        ;
                                        ;
                                        ;-

                                                BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
                                                                                                        T4.5:
037304  104402                                                                                                  TRAP    C$BSUB
                                9$:     TST     KTFLG                           ;CHECK FOR KT11
                                        BEQ     10$                             ;BR, IF NO KT11
                                        JMP     85$                             ;SKIP THIS TEST
                                10$:    JSR     PC,T23RT3                       ;RESTORE PACKET
                                        JSR     PC,T23REST                      ;SET COMMAND PACKET
                                        JSR     PC,T23RT2                       ;SET UP OTHER COMMAND PACKET

                                        ;*************************************************************************
                                        ;
                                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                        ;
                                        ;*************************************************************************

                                                JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
                                                BCS     20$                     ;BR IF INIT WAS OK
                                                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                                MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
                                                ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
037350  104455                                                                                                  TRAP    C$ERDF
037352  000643                                                                                                  .WORD   419
037354  003550                                                                                                  .WORD   SFIERR
037356  011656                                                                                                  .WORD   SFIMSG
                                20$:
                                                MOV     #T23PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS

                                        ;*************************************************************************
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                        ;
                                        ;*************************************************************************

                                                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
                                                BCS     23$                     ;BR, IF COMMAND ISSUED OK
                                                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
037400  104456                                                                                                  TRAP    C$ERHRD
037402  000644                                                                                                  .WORD   420
037404  004754                                                                                                  .WORD   WRTMSG
```

```
     037406 011656                                                       .WORD   SFIMSG

                                   ;*********************************************************
                                   ;
                                   ;WRITE DATA, ACK,CVL=1
                                   ;
                                   ;*********************************************************

     037410 012737 000000 040004  23$:   MOV     #0,T23WB+2              ;HIGH ORDER ADDRESS BITS ETC.
     037416 012737 140005 040000  24$:   MOV     #140005,T23PK3          ;WRITE DATA, ACK,CVC=1
     037424 013701 003076                MOV     FREEHI,R1               ;HIGHEST ADDRESS
     037430 162701 000100                SUB     #100,R1                 ;SET ADDRESS A LITTLE LOWER
     037434 010137 040002                MOV     R1,T23WB                ;LOAD INTO THE PACKET
     037440 012737 175000 040006         MOV     #64000.,T23SZ           ;SET UP BUFFER SIZE (64K BYTES)
     037446 012704 040000                MOV     #T23PK3,R4              ;R4 = POINTER TO PACKET
     037452 010465 177776                MOV     R4,TSDB(R5)             ;ISSUE COMMAND
     037456 004737 017104                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
     037462 016501 000000                MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
     037466 012702 104210                MOV     #SC!NXM!SSR!BIT3,R2     ;SET UP EXPECTED
     037472 020102                       CMP     R1,R2                   ;ARE THEY EQUAL
     037474 001415                       BEQ     80$                     ;BR, IF OK ESP. FUNCTION REJECT
     037476 062737 000001 040004         ADD     #1,T23WB+2              ;START CUTTING THE HIGH ADDRESS BITS DOWN
     037504 022737 000004 040004         CMP     #4,T23WB+2
     037512 001341                       BNE     24$                     ;IF NOT AT ZERO, KEEP TRYING
     037514 004737 020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
     037520                              ERRHRD  ERRNO,T23TMA,PKTSSR     ;TSSR INCORRECT AFTER WRITE COMMAND
     037520 104456                                                       TRAP    C$ERHRD
     037522 000645                                                       .WORD   421
     037524 040315                                                       .WORD   T23TMA
     037526 011670                                                       .WORD   PKTSSR
     037530                       80$:   CKLOOP                          ;LOOP IF SELECTED
     037530 104406                                                       TRAP    C$CLP1
     037532 004737 041360                JSR     PC,T23RT2               ;CLEAN UP PACKET
     037536 004737 041422                JSR     PC,T23RT3               ;RESTORE PACKET

                                   ;*********************************************************
                                   ;
                                   ;       ISSUE A CALL TO REWIND ROUTINE WHICH WILL WAIT QUITE SOME TIME FOR SSR
                                   ;
                                   ;*********************************************************
     037542 004737 010424                JSR     PC,REWIND               ;CALL THE TAPE REWIND



     037546 012737 102010 037760         MOV     #102010,T23PK2          ;REWIND (POSITION) COMMAND
     037554 012704 037760                MOV     #T23PK2,R4              ;LOAD R4 WITH PACKET ADDRESS
     037560 010465 177776                MOV     R4,TSDB(R5)             ;ISSUE REWIND COMMAND
     037564 004737 017220                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
     037570 103407                       BCS     85$                     ;BR, IF TSSR IS OK (GOOD)
     037572 010001                       MOV     R0,R1                   ;SAVE TSSR CONTENTS
     037574 004737 020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
     037600                              ERRHRD  ERRNO,T23RWN,PKTSSR     ;TSSR IS INCORRECT AFTER REWIND
     037600 104456                                                       TRAP    C$ERHRD
     037602 000646                                                       .WORD   422
     037604 040401                                                       .WORD   T23RWN
     037606 011670                                                       .WORD   PKTSSR
     037610                       85$:
```

```
 6459 037610                          SKIP:
 6460 037610                          130$:
 6461 037610                                ENDSUB                    ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
      037610                                                                L10055:
      037610  104403                                                              TRAP    C$ESUB
 6462 037612  023727  002170  000031        CMP     FATFLG,#25.               ;IS ERROR COUNT AT 25
 6463 037620  002402                        BLT     999$                      ;BR, IF LESS THAN 25
 6464 037622  004737  020136                JSR     PC,CKDROP                 ;TRY TO DROP THE UNIT
 6465 037626                          999$:
 6466 037626  004737  017340                JSR     PC,TSTLOOP          ;DO WE NEED TO ITERATE TEST
 6467 037632  103002                        BCC     163$                ;BR, IF NO LOOP REQUIRED
 6468 037634  000137  035566                JMP     T23LOOP             ;EXECUTE AGAIN
 6469 037640                          163$:
 6470 037640                                EXIT    TST                 ;ALL DONE THIS TEST
      037640  104432                                                              TRAP    C$EXIT
      037642  001604                                                              .WORD   L10050-.
```

```
 6472                                   ;+
 6473                                   ;LOCAL STORAGE FOR THIS TEST
 6474                                   ;-
 6476 037644                                    .BLKB   10-<.-TUV2A87>
 6478 037650                            T23PACKET:                      ;COMMAND PACKET FOR TEST
 6479 037650  100004                            .WORD   100004          ;WRITE CHARACTERISTICS COMMAND, WITH ACK
 6480 037652  037660                            .WORD   T23DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
 6481 037654  000000                            .WORD   0
 6482 037656  000010                            .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
 6483 037660                            T23DATA:                        ;CHARACTERISTICS DATA BLOCK
 6484 037660  037670                            .WORD   T23BFR          ;ADDRESS OF MESSAGE BUFFER
 6485 037662  000000                            .WORD   0
 6486 037664  000012                            .WORD   10.             ;LENGTH OF MESSAGE BUFFER
 6487 037666  000000                            .WORD   0
 6488 037670                            T23BFR: .BLKW   25.             ;MESSAGE BUFFER
 6489                                   ;
 6490                                   ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
 6491                                   ;
 6493 037752                                    .BLKB   10-<.-TUV2A87>
 6495 037760                            T23PK2:
 6496 037760  100006                            .WORD   100006          ;WRITE SUB SYS MEM COMMAND, AND ACK
 6497 037762  040012                            .WORD   T23BF2          ;ADDRESS OF SELECT BLOCK DATA
 6498 037764  000000                            .WORD   0
 6499 037766  000006                            .WORD   6.              ;SIZE OF DATA PACKET
 6500
 6502 037770                                    .BLKB   10-<.-TUV2A87>
 6504 040000                            T23PK3:
 6505 040000  100005                            .WORD   100005          ;WRITE COMMAND,  AND ACK
 6506 040002  000000                    T23WB:  .WORD   0               ;ADDRESS OF WRITE BUFFER
 6507 040004  000000                            .WORD   0
 6508 040006  000000                    T23SZ:  .WORD   0               ;SIZE OF BUFFER (EXTENT)
 6509                                           .EVEN
 6510                                   ;
 6511 040010  000000                    T23RSZ: .WORD   0               ;LARGEST TAPE RECORD IN BYTES
 6512                                   ;
 6513                                   ;
 6514 040012                            T23BF2:
 6515 040012     010                    T23BS0: .BYTE   10              ;BSEL0 AREA
 6516 040013     200                    T23BS1: .BYTE   200             ;BSEL1 AREA
 6517 040014  000000                    T23S2:  .WORD   0               ;SEL 2 AREA
 6518 040016  000000                    T23S3:  .WORD   0               ;DATA AREA
 6519                                   ;
 6520                                   ;
 6521                                           .EVEN
 6522                                   ;TAPE MOTION PACKET COMMAND VALUES
 6523
 6524 040020  100005                    T23WD:  .WORD   100005          ;WRITE DATA (NEXT)
 6525 040022  100405                    T23WDR: .WORD   100405          ;WRITE DATA RETRY
 6526 040024  102005                    T23CON: .WORD   102005          ;WRITE CONTINOUS
 6527 040026  177777                            .WORD   177777          ;END OF DATA
 6528
 6529
```

```
6531
6532                                    ;+
6533                                    ;LOCAL TEXT MESSAGES FOR TEST
6534                                    ;-
6535
6536 040030  127  122  111  T23SSR: .ASCIZ  'WRITE Command Not Accepted'
6537 040063  105  117  124  T23ET:  .ASCIZ  'EOT Not Found In 12000 4k Writes, (Use Shorter Tape)'
6538 040150  127  122  111  T23EOT: .ASCIZ  'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
6539 040226  124  123  123  T23TM:  .ASCIZ  'TSSR Not Correct After WRITE Command Reject Due To NXM'
6540 040315  124  123  123  T23TMA: .ASCIZ  'TSSR Not Correct After WRITE To Non-Existent Memory'
6541 040401  122  145  167  T23RWN: .ASCIZ  'Rewind (POSITION) Command Not Accepted'
6542 040450  122  101  115  T23RNC: .ASCIZ  'RAM Error, Correct Data Pattern Not In Ram'
6543 040523  124  123  123  T23AM3: .ASCIZ  'TSSR Init. Failed After WRITE Command'
6544 040571  104  162  151  T23OFL: .ASCIZ  'Drive 7 Select Failed To Set "OFL" In TSSR'
6545 040644  124  123  123  T23WDD: .ASCIZ  'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
6546 040733  124  123  123  T23WDC: .ASCIZ  'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
6547 041035  103  126  103  T23VCK: .ASCIZ  'CVC Set, Didn't Reset VCK In Message Buffer'
6548 041110  124  123  102  T23BA:  .ASCIZ  'TSBA Not Correct After WRITE DATA Command'
6549 041162  127  122  111  T23WSS: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
6550 041251  102  141  163  TST23ID:        .ASCIZ  'Basic Write'
6551                                .EVEN
6552                                    ;+
6553                                    ;
6554                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
6555                                    ;WRITE SUBSYSTEM MEMORY COMMAND
6556                                    ;
6557                                    ;-
6558
6559 041266                         T23REST:
6560 041266                                SAVREG                  ;SAVE THE REGISTERS
6561 041272  012701  037650                MOV     #T23PACKET,R1   ;START OF THE PACKET
6562 041276  012721  100004                MOV     #100004,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK
6563 041302  012721  037660                MOV     #T23DATA,(R1)+  ;ADDRESS OF CHARAISTICS DATA BLOCK
6564 041306  005021                        CLR     (R1)+           ;EXTENDED ADDRESS
6565 041310  012721  000012                MOV     #10.,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
6566 041314  012721  037670                MOV     #T23BFR,(R1)+   ;ADDRESS OF MESSAGE BUFFER
6567 041320  005021                        CLR     (R1)+
6568 041322  012721  000024                MOV     #20.,(R1)+      ;LENGTH OF MESSAGE BUFFER
6569 041326  005021                        CLR     (R1)+
6570 041330  012711  000000                MOV     #0,(R1)         ;SELECT DRIVE ZERO
6571 041334  012702  000030                MOV     #24.,R2         ;NUMBER OF LOCATIONS TO BE CLEARED
6572 041340  012762  177777  037670  64$:  MOV     #177777,T23BFR(R2) ;ALL ONES TO MESSAGE BUFFER
6573 041346  005742                        TST     -(R2)           ;BUMP DOWN TO NEXT LOCATION
6574 041350  020227  000000                CMP     R2,#0           ;R2 AT ZERO YET
6575 041354  001371                        BNE     64$             ;KEEP GOING UNTIL DONE
6576 041356  000207                        RTS     PC              ;RETURN
6577
6578
6579 041360                         T23RT2:
6580 041360                                SAVREG                  ;SAVE THE REGISTERS
6581 041364  012701  037760                MOV     #T23PK2,R1      ;START OF THE PACKET
6582 041370  012721  100006                MOV     #100006,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK
6583 041374  012721  040012                MOV     #T23BF2,(R1)+   ;ADDRESS OF DATA BLOCK
6584 041400  005021                        CLR     (R1)+           ;EXTENDED ADDRESS
6585 041402  012721  000006                MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
6586 041406  012701  040012                MOV     #T23BF2,R1      ;POINT TO DATA SEL AREA
6587 041412  005021                        CLR     (R1)+
```

```
6588 041414  005021                        CLR     (R1)+
6589 041416  005011                        CLR     (R1)
6590 041420  000207                        RTS     PC                      ;RETURN
6591 041422                          T23RT3:
6592 041422                                SAVREG                          ;SAVE THE REGISTERS
6593 041426  012701  040000                MOV     #T23PK3,R1              ;START OF THE PACKET
6594 041432  012721  100005                MOV     #100005,(R1)+           ;WRITE  TAPE. WITH ACK
6595 041436  005021                        CLR     (R1)+                   ;ADDRESS OF DATA BLOCK
6596 041440  005021                        CLR     (R1)+                   ;EXTENDED ADDRESS
6597 041442  005011                        CLR     (R1)                    ;SIZE OF DATA BLOCK
6598 041444  000207                        RTS     PC                              ;RETURN
6599 041446                                ENDTST
     041446                                                          L10050:
     041446  104401                                                        TRAP    C$ETST
```

```
                                        .SBTTL  TEST  5: BASIC READ DATA (FORWARD AND REVERSE)
                                ;+
                                ;
                                ;THIS TEST VERIFIES THAT THE READ FORWARD AND READ REVERSE
                                ;COMMANDS OPERATE PROPERLY. VARIOUS COMBINATIONS OF ODD AND EVEN
                                ;DATA BUFFER BOUNDARIES, RECORD SIZES (UP TO 64K BYTES IF MEMORY
                                ;SPACE IS AVAILIABLE), AND BYTE-SWAP CONTROL ARE USED. THIS TEST
                                ;OF COURSE, FURTHER VERIFIES THE WRITE DATA COMMAND BY ACTUALLY
                                ;READING AND VERIFYING WRITTEN DATA. ALSO TESTED ARE PROPER
                                ;TERMINATIONS ON EXCEPTIONAL OR ERROR CONDITIONS: RECORD LENGTH
                                ;LONG, RECORD LENGTH SHORT, READ REVERSE AT BOT, ILLEGAL DATA
                                ;BUFFER ADDRESSES, ILLEGAL CODES IN THE MODE FIELD OF THE BASIC
                                ;READ COMMAND, AND DATA BUFFERS IN NON-EXISTANT MEMORY. THE TEST
                                ;
                                ;
                                ;THE TEST CONSISTS OF THE FOLLOWING 12 SUBTESTS
                                ;
                                ;
                                ;
                                ;-
                                        BGNTST
041450                                                                          T5::
041450
041450  005037  002170                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
041454  005037  003100                  CLR     KTFLG           ;HOLD OFF KT11
041460  012737  005755  002146          MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY ERROR MESSAGE
041466  005037  003102                  CLR     KTENABLE                ;TURN OFF KT11
041472  004737  020230                  JSR     PC,KTOFF                ;TURN KT11 OFF
041476  012700  052462                  MOV     #TST24ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
041502  004737  017372                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
041506  012737  000001  002164          MOV     #1,LOOPCNT              ;PERFORM 1 ITERATIONS
                                ;+
041514                          T24LOOP:
                                ;
```

```
                                        ;+
                                        ;
                                        ;TEST 5, SUBTEST 1
                                        ;
                                        ;VERIFIES THAT READ FORWARD COMMANDS WITH SWB=0
                                        ;OPERATES PROPERLY. THE TAPE IS FIRST REWOUND AND THEN
                                        ;WRITTEN WITH A SERIES OF TEST RECORDS VARING IN
                                        ;LENGTH AND DATA CONTENT. THE TAPE IS THEN REWOUND
                                        ;AGAIN AND THE RECORD READ SEQUENTIALLY AND RESULTS
                                        ;(STATUS, DATA, ETC.) VERIFIED. THE BYTE COUNT ON
                                        ;EACH READ FORWARD COMMAND IS SET TO THE LENGTH OF THE
                                        ;EXPECTED RECORD, SO NO EXCEPTIONAL CONDITIONS SHOULD
                                        ;OCCUR.
                                        ;
                                        ;
                                        ;
                                        ;-
041514                                          BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
                                                                                        T5.1:
041514  104402                                                                          TRAP    C$BSUB
041516  004737  052664                        JSR     PC,T24RT3               ;SET UP OTHER COMMAND PACKET
041522  004737  052530                        JSR     PC,T24REST              ;SET COMMAND PACKET
041526  004737  052622                        JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET
041532                                          RFLAGS  R0                      ;READ SUPERVISOR FLAGS
041532  104421                                                                  TRAP    C$RFLA
041534  032700  000100                        BIT     #BIT6,R0                ;CHECK FOR THE ILS BIT SET
041540  001002                                BNE     9$                      ;IF SET SKIP THE PATCH
041542  000137  042266                        JMP     192$                    ;PATCH IT OUT
041546                                  9$:
                                        ;*************************************************************************
                                        ;
                                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                        ;
                                        ;*************************************************************************

041546  004737  016630                        JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
041552  103407                                BCS     20$                     ;BR IF INIT WAS OK
041554  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
041560  010001                                MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
041562                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
041562  104455                                                                  TRAP    C$ERDF
041564  000765                                                                  .WORD   501
041566  003550                                                                  .WORD   SFIERR
041570  011656                                                                  .WORD   SFIMSG
041572                                  20$:
041572  012704  050040                        MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
                                        ;*************************************************************************
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                        ;
                                        ;*************************************************************************

041576  004737  010322                        JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
041602  103407                                BCS     24$                     ;BR, IF COMMAND ISSUED OK
041604  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
041610  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
6695 041612                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     041612  104456                                                                TRAP    C$ERHRD
     041614  000766                                                                .WORD   502
     041616  004754                                                                .WORD   WRTMSG
     041620  011656                                                                .WORD   SFIMSG
6696 041622                     24$:    CKLOOP                          ;LOOP IF SELECTED
     041622  104406                                                                TRAP    C$CLP1
6697
6698                            ;*************************************************************************
6699                            ;
6700                            ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6701                            ;
6702                            ;*************************************************************************
6703
6704 041624  004737  010424              JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
6705 041630  103407                      BCS     30$                     ;BR, IF NO PROBLEM
6706 041632  010001                      MOV     R0,R1                   ;SAVE TSSR
6707 041634  004737  020064              JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
6711 041640                              ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
     041640  104456                                                                TRAP    C$ERHRD
     041642  000767                                                                .WORD   503
     041644  051326                                                                .WORD   T24RWN
     041646  011670                                                                .WORD   PKTSSR
6712 041650                     30$:    CKLOOP                          ;LOOP IF SELECTED
     041650  104406                                                                TRAP    C$CLP1
6713
6714                            ;*************************************************************************
6715                            ;
6716                            ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6717                            ;
6718                            ;*************************************************************************
6719
6720 041652  013701  050066              MOV     T24BFR+6,R1             ;PICK UP XST0
6721 041656  010102                      MOV     R1,R2                   ;SET UP EXPECTED
6722 041660  052702  000002              BIS     #BIT1,R2                ;SET BOT BIT IN EXPECTED
6723 041664  020102                      CMP     R1,R2                   ;DOES EXP = REC'D
6724 041666  001406                      BEQ     40$                     ;BR, IF EQUAL (OK)
6725 041670  004737  020064              JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
6729 041674                              ERRHRD  ERRNO,T24BOT,EXPREC     ;TAPE NOT AT BOT AFTER REWIND
     041674  104456                                                                TRAP    C$ERHRD
     041676  000770                                                                .WORD   504
     041700  051043                                                                .WORD   T24BOT
     041702  016330                                                                .WORD   EXPREC
6730 041704                     40$:    CKLOOP                          ;LOOP IF SELECTED
     041704  104406                                                                TRAP    C$CLP1
6731 041706  012703  000400              MOV     #256.,R3                ;RECORD SIZE
6732 041712  013737  003072  050172      MOV     FREE,T24RB              ;STARTING WRITE BUFFER ADDRESS
6733
6734                            ;*************************************************************************
6735                            ;
6736                            ;WRITE DATA,CVC=1,ACK COMMAND
6737                            ;
6738                            ;*************************************************************************
6739
6740 041720  012737  140005  050170      MOV     #140005,T24PK3          ;WRITE DATA,CVC=1,ACK COMMAND
6741 041726  012704  050170              MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
6742 041732                     65$:
```

```
6743 041732  010300                    MOV     R3,R0                   ;SET PATTERN IN CORRECT REGISTER
6744 041734  004737  020356             JSR     PC,FILLMEM              ;FILL MEMORY WITH RECORD SIZE
6745 041740  010337  050176             MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
6746 041744  010465  177776             MOV     R4,TSDB(R5)             ;ISSUE COMMAND
6747 041750  004737  017104             JSR     PC,WAITF                ;WAIT FOR SSR TO SET
6748 041754  016501  000000             MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
6749 041760  012702  000200             MOV     #SSR,R2                 ;SET UP EXPECTED
6750 041764  020102                    CMP     R1,R2                   ;ARE THEY EQUAL
6751 041766  001406                    BEQ     75$                     ;BR, IF OK
6752 041770  004737  020064             JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
6756                                                                    ;SOFT ERROR, REALLY CHECKING THE
6757                                                                    ;READ COMMAND
6758 041774                            ERRSOFT ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER WRITE DATA
     041774  104457                                                                    TRAP    C$ERSOFT
     041776  000771                                                                    .WORD   505
     042000  005011                                                                    .WORD   WRTERR
     042002  011670                                                                    .WORD   PKTSSR
6759 042004                    75$:    CKLOOP                          ;LOOP IF SELECTED
     042004  104406                                                                    TRAP    C$CLP1
6760 042006  005723                    TST     (R3)+           ;BUMP RECORD SIZE
6761 042010  022703  000414             CMP     #268.,R3                ;END OF RECORD YET
6762 042014  001346                    BNE     65$                     ;BR, IF MORE RECORDS TO WRITE
6763 042016                    80$:    CKLOOP                          ;LOOP IF SELECTED
     042016  104406                                                                    TRAP    C$CLP1
6764 042020                    120$:
6765
6766                           ;*************************************************************************
6767                           ;
6768                           ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6769                           ;
6770                           ;*************************************************************************
6771
6772 042020  012703  000012             MOV     #10.,R3                 ;SPECIAL MULTIPLE REWIND
6773 042024  004737  010424     125$:   JSR     PC,REWIND               ;ISSUE REWIND COMMAND
6774 042030  103407                    BCS     130$                    ;BR, IF NO PROBLEM
6775 042032  010001                    MOV     R0,R1                   ;SAVE TSSR
6776 042034  004737  020064             JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
6780 042040                            ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
     042040  104456                                                                    TRAP    C$ERHRD
     042042  000772                                                                    .WORD   506
     042044  051326                                                                    .WORD   T24RWN
     042046  011670                                                                    .WORD   PKTSSR
6781 042050                    130$:   CKLOOP                          ;LOOP IF SELECTED
     042050  104406                                                                    TRAP    C$CLP1
6782
6783                           ;*************************************************************************
6784                           ;
6785                           ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6786                           ;
6787                           ;*************************************************************************
6788
6789 042052  013701  050066             MOV     T24BFR+6,R1             ;PICK UP XST0
6790 042056  010102                    MOV     R1,R2                   ;SET UP EXPECTED
6791 042060  052702  000002             BIS     #BIT1,R2                ;SET EOT BIT IN EXPECTED
6792 042064  020102                    CMP     R1,R2                   ;DOES EXP = REC'D
6793 042066  001407                    BEQ     140$                    ;BR, IF EQUAL (OK)
6794 042070  077323                    SOB     R3,125$                 ;DO ANOTHER REWIND BEFORE REPORTING ERROR
```

```
 6795 042072  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6799 042076                                ERRHRD  ERRNO,T24BOT,EXPREC     ;TAPE NOT AT BOT AFTER REWIND
      042076  104456                                                                TRAP    C$ERHRD
      042100  000773                                                                .WORD   507
      042102  051043                                                                .WORD   T24BOT
      042104  016330                                                                .WORD   EXPREC
 6800 042106                        140$:   CKLOOP                          ;LOOP IF SELECTED
      042106  104406                                                                TRAP    C$CLP1
 6801 042110  012703  000400                MOV     #256.,R3                ;RECORD SIZE
 6802 042114  013737  003072  050172        MOV     FREE,T24RB              ;STARTING READ BUFFER ADDRESS
 6803
 6804                               ;*****************************************************************
 6805                               ;
 6806                               ;READ DATA,CVC=1,ACK COMMAND
 6807                               ;
 6808                               ;*****************************************************************
 6809 042122  012737  140001  050170        MOV     #140001,T24PK3          ;READ DATA,CVC=1,ACK COMMAND
 6810 042130  012704  050170        165$:   MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
 6811 042134  010337  050176                MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
 6812 042140  010465  177776                MOV     R4,TSDB(R5)             ;ISSUE COMMAND
 6813 042144  004737  017104                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 6814 042150  016501  000000                MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
 6815 042154  012702  000200                MOV     #SSR,R2                 ;SET UP EXPECTED
 6816 042160  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
 6817 042162  001406                        BEQ     170$                    ;BR, IF OK
 6818 042164  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6822 042170                                ERRHRD  ERRNO,RDERR,PKTSSR      ;TSSR INCORRECT AFTER READ DATA
      042170  104456                                                                TRAP    C$ERHRD
      042172  000774                                                                .WORD   508
      042174  005104                                                                .WORD   RDERR
      042176  011670                                                                .WORD   PKTSSR
 6823 042200                        170$:   CKLOOP                          ;LOOP IF SELECTED
      042200  104406                                                                TRAP    C$CLP1
 6824 042202  013702  003072                MOV     FREE,R2                 ;GET BUFFER ADDRESS
 6825 042206  010304                        MOV     R3,R4                   ;CURRENT RECORD SIZE
 6826 042210  162704  000400                SUB     #256.,R4                ;FIRST LOCATION IN BUFFER
 6827 042214  060204                173$:   ADD     R2,R4                   ;GET LOCATION IN BUFFER (ADDRESS)
 6828 042216  021403                        CMP     (R4),R3                 ;CHECK DATA READ (R3=DATA ALSO)
 6829 042220  001410                        BEQ     180$                    ;BR, IF ALL IS WELL
 6830 042222  011401                        MOV     (R4),R1                 ;RECD DATA
 6831 042224  010302                        MOV     R3,R2                   ;EXPECTED DATA
 6832 042226  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 6836 042232                                ERRHRD  ERRNO,T24DTA,EXPREC     ;DATA READ NOT = WRITTEN
      042232  104456                                                                TRAP    C$ERHRD
      042234  000775                                                                .WORD   509
      042236  051110                                                                .WORD   T24DTA
      042240  016330                                                                .WORD   EXPREC
 6837 042242                        180$:   CKLOOP                          ;LOOP IF SELECTED
      042242  104406                                                                TRAP    C$CLP1
 6838 042244  005724                        TST     (R4)+           ;BUMP TO NEXT LOCATION
 6839 042246  160204                        SUB     R2,R4           ;GET BACK TO CORRECT SIZE
 6840 042250  020403                        CMP     R4,R3                   ;END OF RECORD YET
 6841 042252  001360                        BNE     173$                    ;BR, IF NOT AT END OF RECORD
 6842 042254  005723                        TST     (R3)+                   ;BUMP RECORD SIZE
 6843 042256  022703  000412                CMP     #266.,R3                ;END OF RECORD YET
 6844 042262  001322                        BNE     165$                    ;BR, IF MORE RECORDS TO WRITE
 6845 042264                        190$:   CKLOOP                          ;LOOP IF SELECTED
```

```
        042264  104406                                                          TRAP    C$CLP1
 6846   042266                          192$:   ENDSUB                  ;>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>
        042266                                                                  L10C57:
        042266  104403                                                          TRAP    C$ESUB
 6847   042270  023727  J02170  000031          CMP     FAT+LG,#25.             ;IS ERROR COUNT AT 25
 6848   042276  002402                          BLT     999$                    ;BR, IF LESS THAN 25
 6849   042300  004737  020136                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 6850   042304                          999$:
```

```
                                ;+
                                ;
                                ;TEST 5, SUBTEST 2
                                ;
                                ;VERIFIES THAT READ DATA COMMANDS WITH CVC=1 AND THE
                                ;SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST
                                ;SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2.
                                ;THE RESULTS, EXCEPT FOR RAM CONTENTS, SHOULD BE THE SAME.
                                ;
                                ;
                                ;-
042304                                  BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
042304                                                                                  T5.2:
042304  104402                                                                          TRAP    C$BSUB
042306  004737  052664                JSR     PC,T24RT3               ;SET UP OTHER COMMAND PACKET
042312  004737  052530                JSR     PC,T24REST              ;SET COMMAND PACKET
042316  004737  052622                JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET

                                ;*************************************************************************
                                ;
                                ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                ;
                                ;*************************************************************************

042322  004737  016630                JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
042326  103407                        BCS     20$                     ;BR IF INIT WAS OK
042330  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
042334  010001                        MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
042336                                ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
042336  104455                                                                          TRAP    C$ERDF
042340  000776                                                                          .WORD   510
042342  003550                                                                          .WORD   SFIERR
042344  011656                                                                          .WORD   SFIMSG
042346                          20$:
042346  012704  050040                MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS

                                ;*************************************************************************
                                ;
                                ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                ;
                                ;*************************************************************************

042352  004737  010322                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
042356  103407                        BCS     24$                     ;BR, IF COMMAND ISSUED OK
042360  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
042364  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
042366                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
042366  104456                                                                          TRAP    C$ERHRD
042370  000777                                                                          .WORD   511
042372  004754                                                                          .WORD   WRTMSG
042374  011656                                                                          .WORD   SFIMSG
042376                          24$:    CKLOOP                          ;LOOP IF SELECTED
042376  104406                                                                          TRAP    C$CLP1

                                ;*************************************************************************
                                ;
```

```
                                        ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
                                        ;
                                        ;*************************************************************

042400  004737  010424                        JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
042404  103407                                BCS     30$                     ;BR, IF NO PROBLEM
042406  010001                                MOV     R0,R1                   ;SAVE TSSR
042410  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
042414                                        ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
042414  104456                                                                TRAP    C$ERHRD
042416  001000                                                                .WORD   512
042420  051326                                                                .WORD   T24RWN
042422  011670                                                                .WORD   PKTSSR
042424                                  30$:    CKLOOP                          ;LOOP IF SELECTED
042424  104406                                                                TRAP    C$CLP1

                                        ;*************************************************************
                                        ;
                                        ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XSTO)
                                        ;
                                        ;*************************************************************

042426  013701  050066                        MOV     T24BFR+6,R1             ;PICK UP XSTO
042432  010102                                MOV     R1,R2                   ;SET UP EXPECTED
042434  052702  000002                        BIS     #BIT1,R2                ;SET BOT BIT IN EXPECTED
042440  020102                                CMP     R1,R2                   ;DOES EXP = REC'D
042442  001406                                BEQ     40$                     ;BR, IF EQUAL (OK)
042444  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
042450                                        ERRHRD  ERRNO,T24BOT,EXPREC     ;TAPE NOT AT BOT AFTER REWIND
042450  104456                                                                TRAP    C$ERHRD
042452  001001                                                                .WORD   513
042454  051043                                                                .WORD   T24BOT
042456  016330                                                                .WORD   EXPREC
042460                                  40$:    CKLOOP                          ;LOOP IF SELECTED
042460  104406                                                                TRAP    C$CLP1
042462  012703  000400                        MOV     #256.,R3                ;RECORD SIZE
042466  013737  003072  050172                MOV     FREE,T24RB              ;STARTING WRITE BUFFER ADDRESS

                                        ;*************************************************************
                                        ;
                                        ;WRITE DATA,ACK,SWB,CVC=1 COMMAND
                                        ;
                                        ;*************************************************************

042474  012737  150005  050170                MOV     #150005,T24PK3          ;WRITE DATA,ACK,SWB,CVC=1 COMMAND
042502  012704  050170                        MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
042506                                  65$:
042506  010300                                MOV     R3,R0                   ;SET PATTERN IN CORRECT REGISTER
042510  004737  020356                        JSR     PC,FILLMEM              ;FILL MEMORY WITH RECORD SIZE
042514  010337  050176                        MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
042520  010465  177776                        MOV     R4,TSDB(R5)             ;ISSUE COMMAND
042524  004737  017104                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
042530  016501  000000                        MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
042534  012702  000200                        MOV     #SSR,R2                 ;SET UP EXPECTED
042540  020102                                CMP     R1,R2                   ;ARE THEY EQUAL
042542  001406                                BEQ     75$                     ;BR, IF OK
042544  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
```

```
6960                                                    ;SOFT ERROR, REALLY CHECKING THE
6961                                                    ;READ COMMAND
6962 042550                        ERRSOFT ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER WRITE DATA
     042550  104457                                                        TRAP    C$ERSOFT
     042552  001002                                                        .WORD   514
     042554  005011                                                        .WORD   WRTERR
     042556  011670                                                        .WORD   PKTSSR
6963 042560                  75$:  CKLOOP                          ;LOOP IF SELECTED
     042560  104406                                                        TRAP    C$CLP1
6964 042562  005723                  TST     (R3)+                 ;BUMP RECORD SIZE
6965 042564  022703  000414          CMP     #268.,R3              ;END OF RECORD YET
6966 042570  001346                  BNE     65$                   ;BR, IF MORE RECORDS TO WRITE
6967 042572                  80$:  CKLOOP                          ;LOOP IF SELECTED
     042572  104406                                                        TRAP    C$CLP1
6968 042574                  120$:
6969
6970                         ;*************************************************************************
6971                         ;
6972                         ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
6973                         ;
6974                         ;*************************************************************************
6975
6976 042574  004737  010424          JSR     PC,REWIND             ;CALL TAPE REWIND COMMAND
6977 042600  103407                  BCS     130$                  ;BR, IF NO PROBLEM
6978 042602  010001                  MOV     R0,R1                 ;SAVE TSSR
6979 042604  004737  020064          JSR     PC,FATCHK             ;INC AND CHECK FOR MORE THAN 25 ERRORS
6983 042610                          ERRHRD  ERRNO,T24RWN,EXPREC   ;REWIND NOT ACCEPTED
     042610  104456                                                        TRAP    C$ERHRD
     042612  001003                                                        .WORD   515
     042614  051326                                                        .WORD   T24RWN
     042616  016330                                                        .WORD   EXPREC
6984 042620                  130$:  CKLOOP                         ;LOOP IF SELECTED
     042620  104406                                                        TRAP    C$CLP1
6985
6986                         ;*************************************************************************
6987                         ;
6988                         ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
6989                         ;
6990                         ;*************************************************************************
6991
6992 042622  013701  050066          MOV     T24BFR+6,R1           ;PICK UP XST0
6993 042626  010102                  MOV     R1,R2                 ;SET UP EXPECTED
6994 042630  052702  000002          BIS     #BIT1,R2              ;SET BOT BIT IN EXPECTED
6995 042634  020102                  CMP     R1,R2                 ;DOES EXP = REC'D
6996 042636  001406                  BEQ     140$                  ;BR, IF EQUAL (OK)
6997 042640  004737  020064          JSR     PC,FATCHK             ;INC AND CHECK FOR MORE THAN 25 ERRORS
7001 042644                          ERRHRD  ERRNO,T24BOT,EXPREC   ;TAPE NOT AT BOT AFTER REWIND
     042644  104456                                                        TRAP    C$ERHRD
     042646  001004                                                        .WORD   516
     042650  051043                                                        .WORD   T24BOT
     042652  016330                                                        .WORD   EXPREC
7002 042654                  140$:  CKLOOP                         ;LOOP IF SELECTED
     042654  104406                                                        TRAP    C$CLP1
7003 042656  012703  000400          MOV     #256.,R3              ;RECORD SIZE
7004 042662  013737  003072  050172  MOV     FREE,T24RB            ;STARTING READ BUFFER ADDRESS
7005
7006                         ;*************************************************************************
```

```
7007                                    ;
7008                                    ;READ DATA,IE,ACK,SWB COMMAND
7009                                    ;
7010                                    ;*****************************************************************************
7011
7012 042670  012737  110001  050170          MOV     #110001,T24PK3          ;READ DATA,IE,ACK,SWB COMMAND
7013 042676  012704  050170          165$:   MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
7014 042702  010337  050176                  MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
7015 042706  010465  177776                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
7016 042712  004737  017104                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
7017 042716  016501  000000                  MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
7018 042722  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
7019 042726  020102                          CMP     R1,R2                   ;ARE THEY EQUAL
7020 042730  001406                          BEQ     170$                    ;BR, IF OK
7021 042732  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7025 042736                                  ERRHRD  ERRNO,RDERR,PKTSSR      ;TSSR INCORRECT AFTER READ DATA
     042736  104456                                                                  TRAP    C$ERHRD
     042740  001005                                                                  .WORD   517
     042742  005104                                                                  .WORD   RDERR
     042744  011670                                                                  .WORD   PKTSSR
7026 042746                          170$:   CKLOOP                          ;LOOP IF SELECTED
     042746  104406                                                                  TRAP    C$CLP1
7027 042750  013702  003072                  MOV     FREE,R2                 ;GET BUFFER ADDRESS
7028 042754  010304                          MOV     R3,R4                   ;CURRENT RECORD SIZE
7029 042756  162704  000400                  SUB     #256.,R4                ;FIRST LOCATION IN BUFFER
7030 042762  060204                  173$:   ADD     R2,R4                   ;GET LOCATION IN BUFFER (ADDRESS)
7031 042764  021403                          CMP     (R4),R3                 ;CHECK DATA READ (R3=DATA ALSO)
7032 042766  001410                          BEQ     180$                    ;BR, IF ALL IS WELL
7033 042770  011401                          MOV     (R4),R1                 ;RECD DATA
7034 042772  010302                          MOV     R3,R2                   ;EXPECTED DATA
7035 042774  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7039 043000                                  ERRHRD  ERRNO,T24DTA,EXPREC     ;DATA READ NOT = WRITTEN
     043000  104456                                                                  TRAP    C$ERHRD
     043002  001006                                                                  .WORD   518
     043004  051110                                                                  .WORD   T24DTA
     043006  016330                                                                  .WORD   EXPREC
7040 043010                          180$:   CKLOOP                          ;LOOP IF SELECTED
     043010  104406                                                                  TRAP    C$CLP1
7041 043012  005724                          TST     (R4)+                   ;BUMP TO NEXT LOCATION
7042 043014  160204                          SUB     R2,R4                   ;SET SIZE TO CORRECT VALUE
7043 043016  020403                          CMP     R4,R3                   ;END OF RECORD YET
7044 043020  001360                          BNE     173$                    ;BR, IF NOT AT END OF RECORD
7045 043022  005723                          TST     (R3)+                   ;BUMP RECORD SIZE
7046 043024  022703  000412                  CMP     #266.,R3                ;END OF RECORD YET
7047 043030  001322                          BNE     165$                    ;BR, IF MORE RECORDS TO WRITE
7048 043032                          190$:   CKLOOP                          ;LOOP IF SELECTED
     043032  104406                                                                  TRAP    C$CLP1
7049 043034                                  ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
     043034                                                                       L10060:
     043034  104403                                                                  TRAP    C$ESUB
7050 043036  023727  002170  0^0031                  CMP     FATFLG,#25.             ;IS ERROR COUNT AT 25
7051 043044  002402                                  BLT     999$                    ;BR, IF LESS THAN 25
7052 043046  004737  020136                          JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
7053 043052                          999$:
```

```
                                        ;+
                                        ;
                                        ;TEST 5, SUBTEST 3
                                        ;
                                        ;VERIFIES THAT A READ FORWARD COMMAND READING A RECORD
                                        ;LONGER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE
                                        ;STATUS ALERT TERMINATION WITH THE RECORD LENGTH LONG
                                        ;(RLL) BIT SET.
                                        ;
                                        ;
                                        ;-

043052                                          BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
043052                                                                                  T5.3:
043052  104402                                                                          TRAP    C$BSUB
043054  004737  052664                            JSR     PC,T24RT3               ;SET UP OTHER COMMAND PACKET
043060  004737  052530                            JSR     PC,T24REST              ;SET COMMAND PACKET
043064  004737  052622                            JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET

                                        ;*************************************************************************
                                        ;
                                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                        ;
                                        ;*************************************************************************

043070  004737  016630                            JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
043074  103407                                    BCS     20$                     ;BR IF INIT WAS OK
043076  004737  020064                            JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
043102  010001                                    MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
043104                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
043104  104455                                                                          TRAP    C$ERDF
043106  001007                                                                          .WORD   519
043110  003550                                                                          .WORD   SFIERR
043112  011656                                                                          .WORD   SFIMSG
043114                                  20$:
043114  012704  050040                            MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS

                                        ;*************************************************************************
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                        ;
                                        ;*************************************************************************

043120  004737  010322                            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
043124  103407                                    BCS     24$                     ;BR, IF COMMAND ISSUED OK
043126  004737  020064                            JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
043132  010001                                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043134                                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
043134  104456                                                                          TRAP    C$ERHRD
043136  001010                                                                          .WORD   520
043140  004754                                                                          .WORD   WRTMSG
043142  011656                                                                          .WORD   SFIMSG
043144                                  24$:    CKLOOP                          ;LOOP IF SELECTED
043144  104406                                                                          TRAP    C$CLP1

                                        ;*************************************************************************
                                        ;
```

```
                                        ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
                                        ;
                                        ;*************************************************************************

043146  004737  010424                        JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
043152  103407                                BCS     30$                     ;BR, IF NO PROBLEM
043154  010001                                MOV     R0,R1                   ;SAVE TSSR
043156  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
043162                                        ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
043162  104456                                                                TRAP    C$ERHRD
043164  001011                                                                .WORD   521
043166  051326                                                                .WORD   T24RWN
043170  011670                                                                .WORD   PKTSSR
043172                                  30$:    CKLOOP                          ;LOOP IF SELECTED
043172  104406                                                                TRAP    C$CLP1

                                        ;*************************************************************************
                                        ;
                                        ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
                                        ;
                                        ;*************************************************************************

043174  013701  050066                        MOV     T24BFR+6,R1             ;PICK UP XST0
043200  010102                                MOV     R1,R2                   ;SET UP EXPECTED
043202  052702  000002                        BIS     #BIT1,R2                ;SET BOT BIT IN EXPECTED
043206  020102                                CMP     R1,R2                   ;DOES EXP = REC'D
043210  001406                                BEQ     40$                     ;BR, IF EQUAL (OK)
043212  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
043216                                        ERRHRD  ERRNO,T24BOT,EXPREC     ;TAPE NOT AT BOT AFTER REWIND
043216  104456                                                                TRAP    C$ERHRD
043220  001012                                                                .WORD   522
043222  051043                                                                .WORD   T24BOT
043224  016330                                                                .WORD   EXPREC
043226                                  40$:    CKLOOP                          ;LOOP IF SELECTED
043226  104406                                                                TRAP    C$CLP1
043230  012703  001000                        MOV     #512.,R3                ;RECORD SIZE
043234  013737  003072  050172                MOV     FREE,T24RB              ;STARTING WRITE BUFFER ADDRESS

                                        ;*************************************************************************
                                        ;
                                        ;WRITE DATA,ACK,CVC=1 COMMAND
                                        ;
                                        ;*************************************************************************

043242  012737  140005  050170                MOV     #140005,T24PK3          ;WRITE DATA,ACK,CVC=1 COMMAND
043250  012704  050170                        MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
043254                                  65$:
043254  010337  050176                        MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
043260  010465  177776                        MOV     R4,TSDB(R5)             ;ISSUE COMMAND
043264  004737  017104                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
043270  016501  000000                        MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
043274  012702  000200                        MOV     #SSR,R2                 ;SET UP EXPECTED
043300  020102                                CMP     R1,R2                   ;ARE THEY EQUAL
043302  001406                                BEQ     75$                     ;BR, IF OK
043304  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                                                              ;SOFT ERROR, REALLY CHECKING THE
                                                                              ;READ DATA COMMAND
```

```
7163 043310                                ERRSOFT ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER WRITE DATA
     043310  104457                                                                TRAP    C$ERSOFT
     043312  001013                                                                .WORD   523
     043314  005011                                                                .WORD   WRTERR
     043316  011670                                                                .WORD   PKTSSR
7164 043320                          75$:  CKLOOP                                  ;LOOP IF SELECTED
     043320  104406                                                                TRAP    C$CLP1
7165 043322                          120$:
7166
7167                                 ;*************************************************************************
7168                                 ;
7169                                 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7170                                 ;
7171                                 ;*************************************************************************
7172
7173 043322  004737  010424                 JSR     PC,REWIND                     ;CALL TAPE REWIND COMMAND
7174 043326  103407                         BCS     130$                          ;BR, IF NO PROBLEM
7175 043330  010001                         MOV     R0,R1                         ;SAVE TSSR
7176 043332  004737  020064                 JSR     PC,FATCHK                     ;INC AND CHECK FOR MORE THAN 25 ERRORS
7180 043336                                 ERRHRD  ERRNO,T24RWN,PKTSSR           ;REWIND NOT ACCEPTED
     043336  104456                                                                TRAP    C$ERHRD
     043340  001014                                                                .WORD   524
     043342  051326                                                                .WORD   T24RWN
     043344  011670                                                                .WORD   PKTSSR
7181 043346                          130$:  CKLOOP                                 ;LOOP IF SELECTED
     043346  104406                                                                TRAP    C$CLP1
7182
7183                                 ;*************************************************************************
7184                                 ;
7185                                 ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
7186                                 ;
7187                                 ;*************************************************************************
7188
7189 043350  013701  050066                 MOV     T24BFR+6,R1                   ;PICK UP XST0
7190 043354  010102                         MOV     R1,R2                         ;SET UP EXPECTED
7191 043356  052702  000002                 BIS     #BIT1,R2                      ;SET BOT BIT IN EXPECTED
7192 043362  020102                         CMP     R1,R2                         ;DOES EXP = REC'D
7193 043364  001406                         BEQ     140$                          ;BR, IF EQUAL (OK)
7194 043366  004737  020064                 JSR     PC,FATCHK                     ;INC AND CHECK FOR MORE THAN 25 ERRORS
7198 043372                                 ERRHRD  ERRNO,T24BOT,EXPREC           ;TAPE NOT AT BOT AFTER REWIND
     043372  104456                                                                TRAP    C$ERHRD
     043374  001015                                                                .WORD   525
     043376  051043                                                                .WORD   T24BOT
     043400  016330                                                                .WORD   EXPREC
7199 043402                          140$:  CKLOOP                                 ;LOOP IF SELECTED
     043402  104406                                                                TRAP    C$CLP1
7200 043404  012703  000400                 MOV     #256.,R3                      ;RECORD SIZE
7201 043410  013737  003072  050172         MOV     FREE,T24RB                    ;STARTING READ BUFFER ADDRESS
7202
7203                                 ;*************************************************************************
7204                                 ;
7205                                 ;READ DATA,ACK,CVC=1 COMMAND
7206                                 ;
7207                                 ;*************************************************************************
7208
7209 043416  012737  140001  050170         MOV     #140001,T24PK3                ;READ DATA,ACK,CVC=1 COMMAND
7210 043424  012704  050170          165$:  MOV     #T24PK3,R4                    ;SET UP R4 WITH PACKET ADDRESS
```

```
7211 043430  010337  050176                MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
7212 043434  010465  177776                MOV     R4,TSDB(R5)             ;ISSUE COMMAND
7213 043440  004737  017104                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
7214 043444  016501  000000                MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
7215 043450  012702  100204                MOV     #SSR!SC!BIT2,R2         ;SET UP EXPECTED
7216 043454  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
7217 043456  001406                        BEQ     170$                    ;BR, IF OK
7218 043460  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7222 043464                                ERRHRD  ERRNO,T24TRL,PKTSSR     ;TSSR INCORRECT AFTER READ DATA
     043464  104456                                                                TRAP    C$ERHRD
     043466  001016                                                                .WORD   526
     043470  052374                                                                .WORD   T24TRL
     043472  011670                                                                .WORD   PKTSSR
7223 043474                        170$:   CKLOOP                          ;LOOP IF SELECTED
     043474  104406                                                                TRAP    C$CLP1
7224
7225                               ;*************************************************************************
7226                               ;
7227                               ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
7228                               ;
7229                               ;*************************************************************************
7230
7231 043476  013701  050066                MOV     T24BFR+6,R1             ;GET MESSAGE BUFFER
7232 043502  010102                        MOV     R1,R2                   ;SET UP EXPECTED
7233 043504  052702  010000                BIS     #BIT12,R2               ;SET THE RLL BIT IN EXPECTED
7234 043510  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
7235 043512  001406                        BEQ     180$                    ;BR, IF EQUAL (ALL IS WELL)
7236 043514  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7240 043520                                ERRHRD  ERRNO,T24LON,EXPREC     ;THE RLL BIT WAS NOT SET IN XST0
     043520  104456                                                                TRAP    C$ERHRD
     043522  001017                                                                .WORD   527
     043524  052142                                                                .WORD   T24LON
     043526  016330                                                                .WORD   EXPREC
7241 043530                        180$:
7242 043530                                ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
     043530                                                                     L10061:
     043530  104403                                                                TRAP    C$ESUB
7243 043532  023727  002170  000031            CMP     FATFLG,#25.                 ;IS ERROR COUNT AT 25
7244 043540  002402                            BLT     999$                        ;BR, IF LESS THAN 25
7245 043542  004737  020136                    JSR     PC,CKDROP                   ;TRY TO DROP THE UNIT
7246 043546                        999$:
```

```
                                        ;+
                                        ;
                                        ;TEST 5, SUBTEST 4
                                        ;
                                        ;VERIFIES THAT A READ FORWARD COMMAND READING A RECORD
                                        ;SHORTER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE
                                        ;STATUS ALERT TERMINATION WITH THE RECORD LENGTH SHORT
                                        ;(RLS) BITS SET. IT IS VERIFIED THAT THE RESIDUAL BYTE
                                        ;COUNT (RBPCR) IN THE MESSAGE BUFFER CONTAINS THE
                                        ;PROPER NONZERO VALUE (E.G. THE DIFFERENCE BETWEEN
                                        ;THE ORIGINAL BYTE COUNT AND THE ACTUAL RECORD
                                        ;LENGTH).
                                        ;
                                        ;
                                        ;
                                        ;-
                                                BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
                                                                                                T5.4:
043546  104402                                                                                          TRAP    C$BSUB
                                                JSR     PC,T24RT3               ;SET UP OTHER COMMAND PACKET
                                                JSR     PC,T24REST              ;SET COMMAND PACKET
                                                JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET
                                        ;*******************************************************************************
                                        ;
                                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                        ;
                                        ;*******************************************************************************
                                                JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
                                                BCS     20$                     ;BR IF INIT WAS OK
                                                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                                MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
                                                ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
043600  104455                                                                                          TRAP    C$ERDF
043602  001020                                                                                          .WORD   528
043604  003550                                                                                          .WORD   SFIERR
043606  011656                                                                                          .WORD   SFIMSG
                                        20$:
                                                MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
                                        ;*******************************************************************************
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                        ;
                                        ;*******************************************************************************
                                                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
                                                BCS     24$                     ;BR, IF COMMAND ISSUED OK
                                                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
043630  104456                                                                                          TRAP    C$ERHRD
043632  001021                                                                                          .WORD   529
043634  004754                                                                                          .WORD   WRTMSG
043636  011656                                                                                          .WORD   SFIMSG
```

```
7301 043640                          24$:    CKLOOP                          ;LOOP IF SELECTED
     043640  104406                                                                  TRAP    C$CLP1
7302
7303                                 ;*************************************************************************
7304                                 ;
7305                                 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7306                                 ;
7307                                 ;*************************************************************************
7308
7309 043642  004737  010424                  JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
7310 043646  103407                          BCS     30$                     ;BR, IF NO PROBLEM
7311 043650  010001                          MOV     R0,R1                   ;SAVE TSSR
7312 043652  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7316 043656                                  ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
     043656  104456                                                                  TRAP    C$ERHRD
     043660  001022                                                                  .WORD   530
     043662  051326                                                                  .WORD   T24RWN
     043664  011670                                                                  .WORD   PKTSSR
7317 043666                          30$:    CKLOOP                          ;LOOP IF SELECTED
     043666  104406                                                                  TRAP    C$CLP1
7318 043670  012703  000400                  MOV     #256.,R3                ;RECORD SIZE
7319 043674  013737  003072  050172          MOV     FREE,T24RB              ;STARTING WRITE BUFFER ADDRESS
7320
7321                                 ;*************************************************************************
7322                                 ;
7323                                 ;WRITE DATA,ACK,CVC=1 COMMAND
7324                                 ;
7325                                 ;*************************************************************************
7326
7327 043702  012737  140005  050170          MOV     #140005,T24PK3          ;WRITE DATA,ACK,CVC=1 COMMAND
7328 043710  012704  050170                  MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
7329 043714                          65$:
7330 043714  010337  050176                  MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
7331 043720  010465  177776                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
7332 043724  004737  017104                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
7333 043730  016501  000000                  MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
7334 043734  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
7335 043740  020102                          CMP     R1,R2                   ;ARE THEY EQUAL
7336 043742  001406                          BEQ     75$                     ;BR, IF OK
7337 043744  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7341                                                                         ;SOFT ERROR, REALLY CHECKING THE
7342                                                                         ;READ DATA COMMAND
7343 043750                                  ERRSOFT ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER WRITE DATA
     043750  104457                                                                  TRAP    C$ERSOFT
     043752  001023                                                                  .WORD   531
     043754  005011                                                                  .WORD   WRTERR
     043756  011670                                                                  .WORD   PKTSSR
7344 043760                          75$:    CKLOOP                          ;LOOP IF SELECTED
     043760  104406                                                                  TRAP    C$CLP1
7345 043762                          120$:
7346
7347                                 ;*************************************************************************
7348                                 ;
7349                                 ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7350                                 ;
7351                                 ;*************************************************************************
7352
```

```
7353 043762  004737  010424                JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
7354 043766  103407                        BCS     130$                    ;BR, IF NO PROBLEM
7355 043770  010001                        MOV     R0,R1                   ;SAVE TSSR
7356 043772  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7360 043776                                ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
     043776  104456                                                                TRAP    C$ERHRD
     044000  001024                                                                .WORD   532
     044002  051326                                                                .WORD   T24RWN
     044004  011670                                                                .WORD   PKTSSR
7361 044006                        130$:   CKLOOP                          ;LOOP IF SELECTED
     044006  104406                                                                TRAP    C$CLP1
7362 044010  012703  001000                MOV     #512.,R3                ;RECORD SIZE
7363 044014  013737  003072  050172        MOV     FREE,T24RB              ;STARTING READ BUFFER ADDRESS
7364
7365                               ;*************************************************************************
7366                               ;
7367                               ;READ DATA,ACK,CVC=1 COMMAND
7368                               ;
7369                               ;*************************************************************************
7370
7371 044022  012737  140001  050170        MOV     #140001,T24PK3          ;READ DATA,ACK,CVC=1 COMMAND
7372 044030  012704  050170        165$:   MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
7373 044034  010337  050176                MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
7374 044040  010465  177776                MOV     R4,TSDB(R5)             ;ISSUE COMMAND
7375 044044  004737  017104                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
7376 044050  016501  000000                MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
7377 044054  012702  100204                MOV     #SSR!SC!BIT2,R2         ;SET UP EXPECTED
7378 044060  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
7379 044062  001406                        BEQ     170$                    ;BR, IF OK
7380 044064  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7384 044070                                ERRHRD  ERRNO,T24TRL,EXPREC     ;TSSR INCORRECT AFTER READ DATA
     044070  104456                                                                TRAP    C$ERHRD
     044072  001025                                                                .WORD   533
     044074  052374                                                                .WORD   T24TRL
     044076  016330                                                                .WORD   EXPREC
7385 044100                        170$:   CKLOOP                          ;LOOP IF SELECTED
     044100  104406                                                                TRAP    C$CLP1
7386
7387                               ;*************************************************************************
7388                               ;
7389                               ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
7390                               ;
7391                               ;*************************************************************************
7392
7393 044102  013701  050066                MOV     T24BFR+6,R1             ;GET MESSAGE BUFFER
7394 044106  010102                        MOV     R1,R2                   ;SET UP EXPECTED
7395 044110  052702  040000                BIS     #BIT14,R2               ;SET THE RLS BIT IN EXPECTED
7396 044114  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
7397 044116  001406                        BEQ     180$                    ;BR, IF EQUAL (ALL IS WELL)
7398 044120  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7402 044124                                ERRHRD  ERRNO,T24LOP,EXPREC     ;THE RLL BIT WAS NOT SET IN XST0
     044124  104456                                                                TRAP    C$ERHRD
     044126  001026                                                                .WORD   534
     044130  052224                                                                .WORD   T24LOP
     044132  016330                                                                .WORD   EXPREC
7403 044134                        180$:
7404 044134  013701  050064                MOV     T24BFR+4,R1             ;PICK UP RESIDUAL BYTE COUNTER
```

```
 7405 044140  012702  000400                MOV     #256.,R2                ;THIS SHOULD BE THE DIFFERENCE
 7406 044144  020102                        CMP     R1,R2                   ;IS THE DIFFERENCE CORRECT
 7407 044146  001406                        BEQ     190$                    ;BR, IF CORRECT
 7408 044150  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 7412 044154                                ERRHRD  ERRNO,T24PBP,EXPREC     ;RBPCR NOT CORRECT
      044154  104456                                                                TRAP    C$ERHRD
      044156  001027                                                                .WORD   535
      044160  052306                                                                .WORD   T24PBP
      044162  016330                                                                .WORD   EXPREC
 7413 044164                        190$:   CKLOOP                          ;LOOP IF SELECTED
      044164  104406                                                                TRAP    C$CLP1
 7414 044166                                ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
      044166                                                                L10062:
      044166  104403                                                                TRAP    C$ESUB
 7415 044170  023727  002170  000031        CMP     FATFLG,#25.                     ;IS ERROR COUNT AT 25
 7416 044176  002402                        BLT     999$                            ;BR, IF LESS THAN 25
 7417 044200  004737  020136                JSR     PC,CKDROP                       ;TRY TO DROP THE UNIT
 7418 044204                        999$:
```

```
                                        ;+
                                        ;TEST 5, SUBTEST 5
                                        ;VERIFIES THAT READ REVERSE COMMANDS WITH SWB=0
                                        ;OPERATES PROPERLY. THE TAPE IS FIRST REWOUND AND THEN
                                        ;WRITTEN WITH A SERIES OF TEST RECORDS VARING IN
                                        ;LENGTH AND DATA CONTENT. THE TAPE IS THEN READ IN REVERSE
                                        ;SEQUENTIALLY AND THE RESULTS
                                        ;(STATUS, DATA, ETC.) VERIFIED. THE BYTE COUNT ON
                                        ;EACH READ REVERSE COMMAND IS SET TO THE LENGTH OF THE
                                        ;EXPECTED RECORD, SO NO EXCEPTIONAL CONDITIONS SHOULD
                                        ;OCCUR.
                                        ;-
044204                                          BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
044204                                                                                  T5.5:
044204  104402                                                                          TRAP    C$BSUB
044206  004737  052664                          JSR     PC,T24RT3               ;SET UP OTHER COMMAND PACKET
044212  004737  052530                          JSR     PC,T24REST              ;SET COMMAND PACKET
044216  004737  052622                          JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET
                                        ;*************************************************************************
                                        ;
                                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                        ;
                                        ;*************************************************************************
044222  004737  016630                          JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
044226  103407                                  BCS     20$                     ;BR IF INIT WAS OK
044230  004737  020064                          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
044234  010001                                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
044236                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
044236  104455                                                                          TRAP    C$ERLF
044240  001030                                                                          .WORD   536
044242  003550                                                                          .WORD   SFIERR
044244  011656                                                                          .WORD   SFIMSG
044246  012704  050040                  20$:    MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
                                        ;*************************************************************************
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                        ;
                                        ;*************************************************************************
044252  004737  010322                          JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
044256  103407                                  BCS     24$                     ;BR, IF COMMAND ISSUED OK
044260  004737  020064                          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
044264  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
044266                                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
044266  104456                                                                          TRAP    C$ERHRD
044270  001031                                                                          .WORD   537
044272  004754                                                                          .WORD   WRTMSG
044274  011656                                                                          .WORD   SFIMSG
044276                                  24$:    CKLOOP                          ;LOOP IF SELECTED
044276  104406                                                                          TRAP    C$CLP1
                                        ;*************************************************************************
                                        ;
                                        ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
                                        ;
                                        ;*************************************************************************
044300  004737  010424                          JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
044304  103407                                  BCS     30$                     ;BR, IF NO PROBLEM
044306  010001                                  MOV     R0,R1                   ;SAVE TSSR
```

```
7472 044310  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7476 044314                                ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
     044314  104456                                                        TRAP    C$ERHRD
     044316  001032                                                        .WORD   538
     044320  051326                                                        .WORD   T24RWN
     044322  011670                                                        .WORD   PKTSSR
7477 044324                         30$:    CKLOOP                         ;LOOP IF SELECTED
     044324  104406                                                        TRAP    C$CLP1
7478 044326  012703  000400                MOV     #256.,R3                ;RECORD SIZE
7479 044332  013737  003072  050172        MOV     FREE,T24RB              ;STARTING WRITE BUFFER ADDRESS
7480                                ;*************************************************************************
7481                                ;
7482                                ;WRITE DATA,ACK,CVC=1 COMMAND
7483                                ;
7484                                ;*************************************************************************
7485 044340  012737  140005  050170        MOV     #140005,T24PK3          ;WRITE DATA,ACK,CVC=1 COMMAND
7486 044346  012704  050170                MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
7487 044352  010300                 65$:    MOV     R3,R0                   ;SET PATTERN IN CORRECT REGISTER
7488 044354  004737  020356                JSR     PC,FILLMEM              ;FILL MEMORY WITH RECORD SIZE
7489 044360  010337  050176                MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
7490 044364  010465  177776                MOV     R4,TSDB(R5)             ;ISSUE COMMAND
7491 044370  004737  017104                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
7492 044374  016501  000000                MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
7493 044400  012702  000200                MOV     #SSR,R2                 ;SET UP EXPECTED
7494 044404  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
7495 044406  001406                        BEQ     75$                     ;BR, IF OK
7496 044410  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7500                                                                       ;SOFT ERROR, REALLY CHECKING THE
7501                                                                       ;READ DATA COMMAND
7502 044414                                ERRSOFT ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER WRITE DATA
     044414  104457                                                        TRAP    C$ERSOFT
     044416  001033                                                        .WORD   539
     044420  005011                                                        .WORD   WRTERR
     044422  011670                                                        .WORD   PKTSSR
7503 044424                         75$:    CKLOOP                         ;LOOP IF SELECTED
     044424  104406                                                        TRAP    C$CLP1
7504 044426  005723                        TST     (R3)+                   ;BUMP RECORD SIZE
7505 044430  022703  000414                CMP     #268.,R3                ;END OF RECORD YET
7506 044434  001346                        BNE     65$                     ;BR, IF MORE RECORDS TO WRITE
7507 044436                         80$:    CKLOOP                         ;LOOP IF SELECTED
     044436  104406                                                        TRAP    C$CLP1
7508 044440  005743                        TST     -(R3)                   ;SET BACK TO 512.
7509 044442  013737  003072  050172        MOV     FREE,T24RB              ;STARTING READ BUFFER ADDRESS
7510                                ;*************************************************************************
7511                                ;
7512                                ;READ REVERSE DATA,ACK COMMAND
7513                                ;
7514                                ;*************************************************************************
7515 044450  012737  100401  050170        MOV     #100401,T24PK3          ;READ REVERSE DATA,ACK COMMAND
7516 044456  012704  050170         165$:  MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
7517 044462  010337  050176                MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
7518 044466  010465  177776                MOV     R4,TSDB(R5)             ;ISSUE COMMAND
7519 044472  004737  017104                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
7520 044476  016501  000000                MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
7521 044502  012702  000200                MOV     #SSR,R2                 ;SET UP EXPECTED
7522 044506  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
7523 044510  001406                        BEQ     170$                    ;BR, IF OK
```

```
 7524 044512 004737 020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 7528 044516                              ERRHRD  ERRNO,T24WDC,PKTSSR     ;TSSR INCORRECT AFTER READ DATA
      044516 104456                                                                              TRAP    C$ERHRD
      044520 001034                                                                              .WORD   540
      044522 051656                                                                              .WORD   T24WDC
      044524 011670                                                                              .WORD   PKTSSR
 7529 044526                       170$:  CKLOOP                          ;LOOP IF SELECTED
      044526 104406                                                                              TRAP    C$CLP1
 7530 044530 013702 003072                MOV     FREE,R2                 ;GET BUFFER ADDRESS
 7531 044534 010304                       MOV     R3,R4                   ;CURRENT RECORD SIZE
 7532 044536 162704 000400                SUB     #256.,R4                ;FIRST LOCATION IN BUFFER
 7533 044542 060204                173$:  ADD     R2,R4                   ;SET POINTER TO FRAME (WORD)
 7534 044544 021403                       CMP     (R4),R3                 ;CHECK DATA READ (R3=DATA ALSO)
 7535 044546 001410                       BEQ     180$                    ;BR, IF ALL IS WELL
 7536 044550 011401                       MOV     (R4),R1                 ;RECD DATA
 7537 044552 010302                       MOV     R3,R2                   ;EXPECTED DATA
 7538 044554 004737 020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 7542 044560                              ERRHRD  ERRNO,T24DTA,EXPREC     ;DATA READ NOT = WRITTEN
      044560 104456                                                                              TRAP    C$ERHRD
      044562 001035                                                                              .WORD   541
      044564 051110                                                                              .WORD   T24DTA
      044566 016330                                                                              .WORD   EXPREC
 7543 044570                       180$:  CKLOOP                          ;LOOP IF SELECTED
      044570 104406                                                                              TRAP    C$CLP1
 7544 044572 005724                       TST     (R4)+                   ;BUMP TO NEXT LOCATION
 7545 044574 160204                       SUB     R2,R4                   ;GET RID OF BASE ADDRESS
 7546 044576 020403                       CMP     R4,R3                   ;END OF RECORD YET
 7547 044600 001360                       BNE     173$                    ;BR, IF NOT AT END OF RECORD
 7548 044602 005743                       TST     -(R3)                   ;BUMP RECORD SIZE
 7549 044604 022703 000400                CMP     #256.,R3                ;END OF RECORD YET
 7550 044610 001322                       BNE     165$                    ;BR, IF MORE RECORDS TO WRITE
 7551 044612                       190$:  CKLOOP                          ;LOOP IF SELECTED
      044612 104406                                                                              TRAP    C$CLP1
 7552 044614                              ENDSUB                          ;>>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>>
      044614                                                                                L10063:
      044614 104403                                                                              TRAP    C$ESUB
 7553 044616 023727 002170 000031         CMP     FATFLG,#25.                       ;IS ERROR COUNT AT 25
 7554 044624 002402                       BLT     999$                              ;BR, IF LESS THAN 25
 7555 044626 004737 020136                JSR     PC,CKDROP                         ;TRY TO DROP THE UNIT
 7556 044632                       999$:
```

```
                                        ;+
                                        ;
                                        ;TEST 5, SUBTEST 6
                                        ;
                                        ;VERIFIES THAT READ DATA COMMANDS WITH CVC=1 AND THE
                                        ;SWAP BYTES (SWB) BIT SET OPERATES PROPERLY. THE TEST
                                        ;SEQUENCE IS IDENTICAL TO THAT USED IN SUBTEST 2.
                                        ;THE RESULTS, EXCEPT FOR RAM CONTENTS, SHOULD BE THE SAME.
                                        ;
                                        ;
                                        ;
                                        ;-
044632                                          BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
044632                                                                                  T5.6:
044632  104402                                                                          TRAP    C$BSUB
044634  004737  052664                        JSR     PC,T24RT3               ;SET UP OTHER COMMAND PACKET
044640  004737  052530                        JSR     PC,T24REST              ;SET COMMAND PACKET
044644  004737  052622                        JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET

                                        ;*************************************************************************
                                        ;
                                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                        ;
                                        ;*************************************************************************

044650  004737  016630                        JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
044654  103407                                BCS     20$                     ;BR IF INIT WAS OK
044656  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
044662  010001                                MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
044664                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
044664  104455                                                                  TRAP    C$ERDF
044666  001036                                                                  .WORD   542
044670  003550                                                                  .WORD   SFIERR
044672  011656                                                                  .WORD   SFIMSG
044674                                  20$:
044674  012704  050040                        MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS

                                        ;*************************************************************************
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                        ;
                                        ;*************************************************************************

044700  004737  010322                        JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
044704  103407                                BCS     24$                     ;BR, IF COMMAND ISSUED OK
044706  004737  020064                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
044712  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
044714                                        ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
044714  104456                                                                  TRAP    C$ERHRD
044716  001037                                                                  .WORD   543
044720  004754                                                                  .WORD   WRTMSG
044722  011656                                                                  .WORD   SFIMSG
044724                                  24$:    CKLOOP                          ;LOOP IF SELECTED
044724  104406                                                                  TRAP    C$CLP1

                                        ;*************************************************************************
                                        ;
```

```
                                        ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
                                        ;
                                        ;*****************************************************************

 044726  004737  010424                         JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
 044732  103407                                 BCS     30$                     ;BR, IF NO PROBLEM
 044734  010001                                 MOV     R0,R1                   ;SAVE TSSR
 044736  004737  020064                         JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 044742                                         ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
 044742  104456                                                                                 TRAP    C$ERHRD
 044744  001040                                                                                 .WORD   544
 044746  051326                                                                                 .WORD   T24RWN
 044750  011670                                                                                 .WORD   PKTSSR
 044752                                 30$:    CKLOOP                          ;LOOP IF SELECTED
 044752  104406                                                                                 TRAP    C$CLP1
 044754  012703  000400                         MOV     #256.,R3                ;RECORD SIZE
 044760  013737  003072  050172                 MOV     FREE,T24RB              ;STARTING WRITE BUFFER ADDRESS


                                        ;*****************************************************************
                                        ;
                                        ;WRITE DATA,ACK,CVC=1,SWB COMMAND
                                        ;
                                        ;*****************************************************************

 044766  012737  150005  050170                 MOV     #150005,T24PK3          ;WRITE DATA,ACK,CVC=1,SWB COMMAND
 044774  012704  050170                         MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
 045000                                 65$:
 045000  010300                                 MOV     R3,R0                   ;SET PATTERN IN CORRECT REGISTER
 045002  004737  020356                         JSR     PC,FILLMEM              ;FILL MEMORY WITH RECORD SIZE
 045006  010337  050176                         MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
 045012  010465  177776                         MOV     R4,TSDB(R5)             ;ISSUE COMMAND
 045016  004737  017104                         JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 045022  016501  000000                         MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
 045026  012702  000200                         MOV     #SSR,R2                 ;SET UP EXPECTED
 045032  020102                                 CMP     R1,R2                   ;ARE THEY EQUAL
 045034  001406                                 BEQ     75$                     ;BR, IF OK
 045036  004737  020064                         JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                                                                ;SOFT ERROR, REALLY CHECKING THE
                                                                                ;READ DATA COMMAND
 045042                                         ERRSOFT ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER WRITE DATA
 045042  104457                                                                                 TRAP    C$ERSOFT
 045044  001041                                                                                 .WORD   545
 045046  005011                                                                                 .WORD   WRTERR
 045050  011670                                                                                 .WORD   PKTSSR
 045052                                 75$:    CKLOOP                          ;LOOP IF SELECTED
 045052  104406                                                                                 TRAP    C$CLP1
 045054  005723                                 TST     (R3)+                   ;BUMP RECORD SIZE
 045056  022703  000414                         CMP     #268.,R3                ;END OF RECORD YET
 045062  001346                                 BNE     65$                     ;BR, IF MORE RECORDS TO WRITE
 045064                                 80$:    CKLOOP                          ;LOOP IF SELECTED
 045064  104406                                                                                 TRAP    C$CLP1
 045066  005743                                 TST     -(R3)                   ;SET RECORD SIZE BACK TO 512.
 045070  013737  003072  050172                 MOV     FREE,T24RB              ;STARTING READ BUFFER ADDRESS

                                        ;*****************************************************************
                                        ;
                                        ;READ REVERSE DATA,ACK,SWB COMMAND
```

```
7662                                    ;
7663                                    ;*************************************************************************
7664
7665 045076  012737  110401  050170      MOV     #110401,T24PK3          ;READ REVERSE DATA,ACK,SWB COMMAND
7666 045104  012704  050170      165$:   MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
7667 045110  010337  050176              MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
7668 045114  010465  177776              MOV     R4,TSDB(R5)             ;ISSUE COMMAND
7669 045120  004737  017104              JSR     PC,WAITF                ;WAIT FOR SSR TO SET
7670 045124  016501  000000              MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
7671 045130  012702  000200              MOV     #SSR,R2                 ;SET UP EXPECTED
7672 045134  020102                      CMP     R1,R2                   ;ARE THEY EQUAL
7673 045136  001406                      BEQ     170$                    ;BR, IF OK
7674 045140  004737  020064              JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7678 045144                              ERRHRD  ERRNO,T24WDC,EXPREC     ;TSSR INCORRECT AFTER READ DATA
     045144  104456                                                              TRAP    C$ERHRD
     045146  001042                                                              .WORD   546
     045150  051656                                                              .WORD   T24WDC
     045152  016330                                                              .WORD   EXPREC
7679 045154                      170$:   CKLOOP                          ;LOOP IF SELECTED
     045154  104406                                                              TRAP    C$CLP1
7680 045156  013702  003072              MOV     FREE,R2                 ;GET BUFFER ADDRESS
7681 045162  010304                      MOV     R3,R4                   ;CURRENT RECORD SIZE
7682 045164  162704  000400              SUB     #256.,R4                ;FIRST LOCATION IN BUFFER
7683 045170  060204      173$:   ADD     R2,R4                   ;SET POINTER TO FRAME (WORD)
7684 045172  021403                      CMP     (R4),R3                 ;CHECK DATA READ (R3=DATA ALSO)
7685 045174  001410                      BEQ     180$                    ;BR, IF ALL IS WELL
7686 045176  011401                      MOV     (R4),R1                 ;RECD DATA
7687 045200  010302                      MOV     R3,R2                   ;EXPECTED DATA
7688 045202  004737  020064              JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7692 045206                              ERRHRD  ERRNO,T24DTA,EXPREC     ;DATA READ NOT = WRITTEN
     045206  104456                                                              TRAP    C$ERHRD
     045210  001043                                                              .WORD   547
     045212  051110                                                              .WORD   T24DTA
     045214  016330                                                              .WORD   EXPREC
7693 045216                      180$:   CKLOOP                          ;LOOP IF SELECTED
     045216  104406                                                              TRAP    C$CLP1
7694 045220  005724                      TST     (R4)+                   ;BUMP TO NEXT LOCATION
7695 045222  160204                      SUB     R2,R4                   ;GET RID OF BASE ADDRESS
7696 045224  020403                      CMP     R4,R3                   ;END OF RECORD YET
7697 045226  001360                      BNE     173$                    ;BR, IF NOT AT END OF RECORD
7698 045230  005743                      TST     -(R3)                   ;BUMP RECORD SIZE
7699 045232  022703  000400              CMP     #256.,R3                ;END OF RECORD YET
7700 045236  001322                      BNE     165$                    ;BR, IF MORE RECORDS TO WRITE
7701 045240                      190$:   CKLOOP                          ;LOOP IF SELECTED
     045240  104406                                                              TRAP    C$CLP1
7702 045242                              ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
     045242                                                                  L10064:
     045242  104403                                                              TRAP    C$ESUB
7703 045244  023727  002170  000031          CMP     FATFLG,#25.                     ;IS ERROR COUNT AT 25
7704 045252  002402                          BLT     999$                            ;BR, IF LESS THAN 25
7705 045254  004737  020136                  JSR     PC,CKDROP                       ;TRY TO DROP THE UNIT
7706 045260                      999$:
```

```
                                        ;+
                                        ;
                                        ;TEST 5, SUBTEST 7
                                        ;
                                        ;VERIFIES THAT A READ REVERSE COMMAND READING A RECORD
                                        ;LONGER THAN THE SPECIFIED BYTE COUNT CAUSES TAPE
                                        ;STATUS ALERT TERMINATION WITH THE RECORD LENGTH LONG
                                        ;(RLL) BIT SET.
                                        ;
                                        ;
                                        ;-

045260                                          BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
045260                                                                                  T5.7:
045260  104402                                                                          TRAP    C$BSUB
045262  004737  052664                          JSR     PC,T24RT3               ;SET UP OTHER COMMAND PACKET
045266  004737  052530                          JSR     PC,T24REST              ;SET COMMAND PACKET
045272  004737  052622                          JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET

                                        ;*************************************************************************
                                        ;
                                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                        ;
                                        ;*************************************************************************

045276  004737  016630                          JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
045302  103407                                  BCS     20$                     ;BR IF INIT WAS OK
045304  004737  020064                          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
045310  010001                                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
045312                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
045312  104455                                                                          TRAP    C$ERDF
045314  001044                                                                          .WORD   548
045316  003550                                                                          .WORD   SFIERR
045320  011656                                                                          .WORD   SFIMSG
045322                                  20$:
045322  012704  050040                          MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS

                                        ;*************************************************************************
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                        ;
                                        ;*************************************************************************

045326  004737  010322                          JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
045332  103407                                  BCS     24$                     ;BR, IF COMMAND ISSUED OK
045334  004737  020064                          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
045340  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
045342                                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
045342  104456                                                                          TRAP    C$ERHRD
045344  001045                                                                          .WORD   549
045346  004754                                                                          .WORD   WRTMSG
045350  011656                                                                          .WORD   SFIMSG
045352                                  24$:    CKLOOP                          ;LOOP IF SELECTED
045352  104406                                                                          TRAP    C$CLP1

                                        ;*************************************************************************
                                        ;
```

```
                                        ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
                                        ;
                                        ;*************************************************************************
                                        
045354  004737  010424                          JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
045360  004737  017220                          JSR     PC,CHKTSSR              ;SEE HOW TSSR IS
045364  103407                                  BCS     30$                     ;BR, IF NO PROBLEM
045366  010001                                  MOV     R0,R1                   ;SAVE TSSR
045370  004737  020064                          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
045374                                          ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
045374  104456                                                                          TRAP    C$ERHRD
045376  001046                                                                          .WORD   550
045400  051326                                                                          .WORD   T24RWN
045402  011670                                                                          .WORD   PKTSSR
045404                                  30$:    CKLOOP                          ;LOOP IF SELECTED
045404  104406                                                                          TRAP    C$CLP1
045406  012703  001000                          MOV     #512.,R3                ;RECORD SIZE
045412  013737  003072  050172                  MOV     FREE,T24RB              ;STARTING WRITE BUFFER ADDRESS
                                        
                                        ;*************************************************************************
                                        ;
                                        ;WRITE DATA,ACK,CVC=1 COMMAND
                                        ;
                                        ;*************************************************************************
                                        
045420  012737  140005  050170                  MOV     #140005,T24PK3          ;WRITE DATA,ACK,CVC=1 COMMAND
045426  012704  050170                          MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
045432                                  65$:
045432  010337  050176                          MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
045436  010465  177776                          MOV     R4,TSDB(R5)             ;ISSUE COMMAND
045442  004737  017104                          JSR     PC,WAITF                ;WAIT FOR SSR TO SET
045446  016501  000000                          MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
045452  012702  000200                          MOV     #SSR,R2                 ;SET UP EXPECTED
045456  020102                                  CMP     R1,R2                   ;ARE THEY EQUAL
045460  001406                                  BEQ     75$                     ;BR, IF OK
045462  004737  020064                          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                                                                ;SOFT ERROR, REALLY CHECKING THE
                                                                                ;READ DATA COMMAND
045466                                          ERRSOFT ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER WRITE DATA
045466  104457                                                                          TRAP    C$ERSOFT
045470  001047                                                                          .WORD   551
045472  005011                                                                          .WORD   WRTERR
045474  011670                                                                          .WORD   PKTSSR
045476                                  75$:    CKLOOP                          ;LOOP IF SELECTED
045476  104406                                                                          TRAP    C$CLP1
045500  012703  000400                          MOV     #256.,R3                ;SIZE OF RECORD
045504  013737  003072  050172                  MOV     FREE,T24RB              ;STARTING READ BUFFER ADDRESS
                                        
                                        ;*************************************************************************
                                        ;
                                        ;READ DATA,ACK COMMAND
                                        ;
                                        ;*************************************************************************
                                        
045512  012737  100401  050170                  MOV     #100401,T24PK3          ;READ DATA,ACK COMMAND
045520  012704  050170                  165$:   MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
045524  010337  050176                          MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
```

```
7813 045530  010465  177776                MOV     R4,TSDB(R5)             ;ISSUE COMMAND
7814 045534  004737  017104                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
7815 045540  016501  000000                MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
7816 045544  012702  100204                MOV     #SSR!SC!BIT2,R2         ;SET UP EXPECTED
7817 045550  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
7818 045552  001406                        BEQ     170$                    ;BR, IF OK
7819 045554  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7823 045560                                ERRHRD  ERRNO,T24TRL,EXPREC     ;TSSR INCORRECT AFTER READ DATA
     045560  104456                                                                TRAP    C$ERHRD
     045562  001050                                                                .WORD   552
     045564  052374                                                                .WORD   T24TRL
     045566  016330                                                                .WORD   EXPREC
7824 045570                        170$:   CKLOOP                          ;LOOP IF SELECTED
     045570  104406                                                                TRAP    C$CLP1
7825
7826                               ;*****************************************************************
7827                               ;
7828                               ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
7829                               ;
7830                               ;*****************************************************************
7831
7832 045572  013701  050066                MOV     T24BFR+6,R1             ;GET MESSAGE BUFFER (XST0)
7833 045576  010102                        MOV     R1,R2                   ;SET UP EXPECTED
7834 045600  052702  010000                BIS     #BIT12,R2               ;SET THE RLL BIT IN EXPECTED
7835 045604  020102                        CMP     R1,R2                   ;ARE THEY EQUAL
7836 045606  001406                        BEQ     180$                    ;BR, IF EQUAL (ALL IS WELL)
7837 045610  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7841 045614                                ERRHRD  ERRNO,T24LON,EXPREC     ;THE RLL BIT WAS NOT SET IN XST0
     045614  104456                                                                TRAP    C$ERHRD
     045616  001051                                                                .WORD   553
     045620  052142                                                                .WORD   T24LON
     045622  016330                                                                .WORD   EXPREC
7842 045624                        180$:
7843 045624                                ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
     045624                                                                     L10065:
     045624  104403                                                                TRAP    C$ESUB
7844 045626  023727  002170  000031                CMP     FATFLG,#25.             ;IS ERROR COUNT AT 25
7845 045634  002402                                BLT     999$                    ;BR, IF LESS THAN 25
7846 045636  004737  020136                        JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
7847 045642                        999$:
7848                               ;+
```

```
 7850                                   ;
 7851                                   ;TEST 5, SUBTEST 8
 7852                                   ;
 7853                                   ;VERIFIES THAT A READ REVERSE COMMAND SPECIFYING A DATA
 7854                                   ;BUFFER STARTING IN NONEXISTANT MEMORY TERMINATES WITH
 7855                                   ;THE PROPER ERROR STATUS WITHOUT MOVING TAPE
 7856                                   ;
 7857                                   ;
 7858                                   ;
 7859                                   ;-
 7860
 7861 045642                                BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
      045642                                                                                    T5.8:
      045642  104402                                                                                TRAP    C$BSUB
 7862 045644  012737  000000  050216        MOV     #0,T24DLY               ;ADDRESS BITS FOR LATER
 7863 045652  005737  003100        9$:     TST     KTFLG                   ;CHECK FOR KT11
 7864 045656  001402                        BEQ     10$                     ;BR, IF NO KT11
 7865 045660  000137  046300                JMP     180$                    ;JUMP IF KT11
 7866 045664  004737  052664        10$:    JSR     PC,T24RT3               ;SET UP OTHER COMMAND PACKET
 7867 045670  004737  052530                JSR     PC,T24REST              ;SET COMMAND PACKET
 7868 045674  004737  052622                JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET
 7869
 7870                                   ;*****************************************************************************
 7871                                   ;
 7872                                   ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
 7873                                   ;
 7874                                   ;*****************************************************************************
 7875
 7876 045700  004737  016630                JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
 7877 045704  103407                        BCS     20$                     ;BR IF INIT WAS OK
 7878 045706  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 7882 045712  010001                        MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
 7883 045714                                ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
      045714  104455                                                                           TRAP    C$ERDF
      045716  001052                                                                           .WORD   554
      045720  003550                                                                           .WORD   SFIERR
      045722  011656                                                                           .WORD   SFIMSG
 7884 045724                            20$:
 7885 045724  012704  050040                MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
 7886
 7887                                   ;*****************************************************************************
 7888                                   ;
 7889                                   ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
 7890                                   ;
 7891                                   ;*****************************************************************************
 7892
 7893 045730  004737  010322                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 7894 045734  103407                        BCS     24$                     ;BR, IF COMMAND ISSUED OK
 7895 045736  004737  020064                JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 7899 045742  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7900 045744                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      045744  104456                                                                           TRAP    C$ERHRD
      045746  001053                                                                           .WORD   555
      045750  004754                                                                           .WORD   WRTMSG
      045752  011656                                                                           .WORD   SFIMSG
 7901 045754                            24$:    CKLOOP                          ;LOOP IF SELECTED
      045754  104406                                                                           TRAP    C$CLP1
```

```
7902
7903                                    ;*************************************************************************
7904                                    ;
7905                                    ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
7906                                    ;
7907                                    ;*************************************************************************
7908
7909 045756  004737  010424                    JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
7910 045762  004737  017220                    JSR     PC,CHKTSSR              ;SEE HOW TSSR IS
7911 045766  103407                            BCS     30$                     ;BR, IF NO PROBLEM
7912 045770  010001                            MOV     R0,R1                   ;SAVE TSSR
7913 045772  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7917 045776                                    ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
      045776  104456                                                                    TRAP    C$ERHRD
      046000  001054                                                                    .WORD   556
      046002  051326                                                                    .WORD   T24RWN
      046004  011670                                                                    .WORD   PKTSSR
7918 046006                            30$:    CKLOOP                          ;LOOP IF SELECTED
      046006  104406                                                                    TRAP    C$CLP1
7919 046010  012703  000400                    MOV     #256.,R3                ;RECORD SIZE
7920 046014  013737  003072  050172            MOV     FREE,T24RB              ;STARTING WRITE BUFFER ADDRESS
7921
7922                                    ;*************************************************************************
7923                                    ;
7924                                    ;WRITE DATA,ACK,CVC=1 COMMAND
7925                                    ;
7926                                    ;*************************************************************************
7927
7928 046022  012737  140005  050170            MOV     #140005,T24PK3          ;WRITE DATA,ACK,CVC=1 COMMAND
7929 046030  012704  050170                    MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
7930 046034                            65$:
7931 046034  010337  050176                    MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
7932 046040  010465  177776                    MOV     R4,TSDB(R5)             ;ISSUE COMMAND
7933 046044  004737  017104                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
7934 046050  016501  000000                    MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
7935 046054  012702  000200                    MOV     #SSR,R2                 ;SET UP EXPECTED
7936 046060  020102                            CMP     R1,R2                   ;ARE THEY EQUAL
7937 046062  001406                            BEQ     75$                     ;BR, IF OK
7938 046064  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7942 046070                                    ERRSOFT ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER WRITE DATA
      046070  104457                                                                    TRAP    C$ERSOFT
      046072  001055                                                                    .WORD   557
      046074  005011                                                                    .WORD   WRTERR
      046076  011670                                                                    .WORD   PKTSSR
7943 046100                            75$:    CKLOOP                          ;LOOP IF SELECTED
      046100  104406                                                                    TRAP    C$CLP1
7944 046102  012703  000400                    MOV     #256.,R3                ;RECORD SIZE
7945 046106  012701  160000                    MOV     #160000,R1              ;START POSSIBLE NXM ADDRESS
7946 046112  012702  177776                    MOV     #177776,R2              ;END POSSIBLE NXM ADDRESS
7947 046116  004737  017260                    JSR     PC,XNXM                 ;CALL NXM FINDER ROUTINE
7948 046122  103402                            BCS     76$                     ;BR IF NXM ADDRESS FOUND
7949 046124  000137  046300                    JMP     180$                    ;JMP OVER CAN'T FIND NXM
7950 046130  010137  050172            76$:    MOV     R1,T24RB                ;STARTING READ REVERSE BUFFER ADDRESS
7951
7952                                    ;*************************************************************************
7953                                    ;
7954                                    ;READ REVERSE DATA,ACK COMMAND
```

```
                                        ;
                                        ;*************************************************************************
                                        ;
046134  013737  050216  050174           MOV     T24DLY,T24RB+2          ;GET BITS 16 AND 17
046142  012737  100401  050170           MOV     #10U401,T24PK3          ;READ REVERSE DATA,ACK COMMAND
046150  012704  050170                   MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
046154  012737  000400  050176           MOV     #256.,T24SZ             ;SET UP RECORD SIZE IN PACKET
046162  010465  177776                   MOV     R4,TSDB(R5)             ;ISSUE COMMAND
046166  004737  017104                   JSR     PC,WAITF                ;WAIT FOR SSR TO SET
046172  016501  000000                   MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
046176  012702  104210                   MOV     #SSR!NXM!SC!BIT3,R2     ;SET UP EXPECTED
046202  020102                           CMP     R1,R2                   ;ARE THEY EQUAL
046204  001417                           BEQ     170$                    ;BR, IF OK
046206  062737  000001  050216           ADD     #1,T24DLY               ;NEXT BUNCH OF MEMORY
046214  022737  000004  050216           CMP     #4,T24DLY               ;TOO MUCH MEMORY
046222  001402                           BEQ     168$                    ;BR IF OVER
046224  000137  045664                   JMP     10$                     ;TRY AGAIN
046230  004737  020064           168$:   JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
046234                                   ERRHRD  ERRNO,T24NXM,PKTSSR     ;TSSR INCORRECT AFTER READ DATA
046234  104456                                                                   TRAP    C$ERHRD
046236  001056                                                                   .WORD   558
046240  050431                                                                   .WORD   T24NXM
046242  011670                                                                   .WORD   PKTSSR
046244                           170$:   CKLOOP                          ;LOOP IF SELECTED
046244  104406                                                                   TRAP    C$LLP1
                                        ;
                                        ;*************************************************************************
                                        ;
                                        ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
                                        ;
                                        ;*************************************************************************
                                        ;
046246  013701  050066                   MOV     T24BFR+6,R1             ;GET MESSAGE BUFFER
046252  010102                           MOV     R1,R2                   ;SET UP EXPECTED
046254  052702  040000                   BIS     #BIT14,R2               ;SET THE RLS BIT IN EXPECTED
046260  020102                           CMP     R1,R2                   ;ARE THEY EQUAL
046262  001406                           BEQ     180$                    ;BR, IF EQUAL (ALL IS WELL)
046264  004737  020064                   JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
046270                                   ERRHRD  ERRNO,T24LOP,EXPREC     ;THE RLL BIT WAS NOT SET IN XST0
046270  104456                                                                   TRAP    C$ERHRD
046272  001057                                                                   .WORD   559
046274  052224                                                                   .WORD   T24LOP
046276  016330                                                                   .WORD   EXPREC
046300                           180$:
                                        ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
046300                                                                           L10066:
046300  104403                                                                   TRAP    C$ESUB
046302  023727  002170  000031                   CMP     FATFLG,#25.             ;IS ERROR COUNT AT 25
046310  002402                                   BLT     999$                    ;BR, IF LESS THAN 25
046312  004737  020136                           JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
046316                           999$:
```

```
                                        ;+
                                        ;
                                        ;TEST 5, SUBTEST 9
                                        ;
                                        ;VERIFIES THAT ILLEGAL BUFFER ADDRESSES CAUSE A
                                        ;FUNCTION REJECT TERMINATION WITH ILLEGAL ADDRESS
                                        ;(ILA) ERROR BIT SET.
                                        ;
                                        ;
                                        ;
                                        ;-
                                        
046316                                          BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
046316                                                                                  T5.9:
046316  104402                                                                          TRAP    C$BSUB
046320  004737  052664                         JSR     PC,T24RT3               ;SET COMMAND PACKET UP CLEAR
046324  004737  052530                         JSR     PC,T24REST              ;SET COMMAND PACKET
046330  004737  052622                         JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET
                                        
                                        ;*************************************************************************
                                        ;
                                        ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                        ;
                                        ;*************************************************************************
                                        
046334  004737  016630                         JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
046340  103407                                 BCS     20$                     ;BR IF INIT WAS OK
046342  004737  020064                         JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
046346  010001                                 MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
046350                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
046350  104455                                                                          TRAP    C$ERDF
046352  001060                                                                          .WORD   560
046354  003550                                                                          .WORD   SFIERR
046356  011656                                                                          .WORD   SFIMSG
046360                                  20$:
046360  012704  050040                         MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
                                        
                                        ;*************************************************************************
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                        ;
                                        ;*************************************************************************
                                        
046364  004737  010322                         JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
046370  103407                                 BCS     24$                     ;BR, IF COMMAND ISSUED OK
046372  004737  020064                         JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
046376  010001                                 MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
046400                                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
046400  104456                                                                          TRAP    C$ERHRD
046402  001061                                                                          .WORD   561
046404  004754                                                                          .WORD   WRTMSG
046406  011656                                                                          .WORD   SFIMSG
046410                                  24$:    CKLOOP                          ;LOOP IF SELECTED
046410  104406                                                                          TRAP    C$CLP1
046412  013737  003072  050172                 MOV     FREE,T24RB              ;ILLEGAL STARTING READ BUFFER ADDRESS
046420  012737  177700  050174                 MOV     #177700,T24RB+2         ;CREATE ILLEGAL ADDRESS
```

```
CZTUXA0 TU80 FRONT END PRT B    MACRO M1200  29-MAR-83 13:32  PAGE 109-1
TEST  5: BASIC READ DATA (FORWARD AND REVERSE)

                                ;*****************************************************************
                                ;
                                ;LEGAL MODE,ACK,CVC=1,READ COMMAND
                                ;
                                ;*****************************************************************

046426  012737  140001  050170          MOV     #140001,T24PK3          ;LEGAL MODE,ACK,CVC=1,READ COMMAND
046434  012704  050170                  MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
046440  012737  000400  050176          MOV     #256.,T24SZ             ;SET UP RECORD SIZE IN PACKET
046446  010465  177776                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
046452  004737  017104                  JSR     PC,WAITF                ;WAIT FOR SSR!BIT1!BIT2 TO SET
046456  016501  000000                  MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
046462  012702  100206                  MOV     #SSR!SC!BIT1!BIT2,R2    ;SET UP EXPECTED
046466  020102                          CMP     R1,R2                   ;ARE THEY EQUAL
046470  001406                          BEQ     75$                     ;BR, IF OK
046472  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
046476                                  ERRHRD  ERRNO,T24WDG,PKTSSR     ;TSSR INCORRECT AFTER READ DATA
046476  104456                                                                  TRAP    C$ERHRD
046500  001062                                                                  .WORD   562
046502  050342                                                                  .WORD   T24WDG
046504  011670                                                                  .WORD   PKTSSR
046506                          75$:    CKLOOP                          ;LOOP IF SELECTED
046506  104406                                                                  TRAP    C$CLP1

                                ;*****************************************************************
                                ;
                                ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
                                ;
                                ;*****************************************************************

046510  013701  050066                  MOV     T24BFR+6,R1             ;GET MESSAGE BUFFER
046514  010102                          MOV     R1,R2                   ;SET UP EXPECTED
046516  052702  000400                  BIS     #BIT8,R2                ;SET THE ILA BIT IN EXPECTED
046522  020102                          CMP     R1,R2                   ;ARE THEY EQUAL
046524  001406                          BEQ     180$                    ;BR, IF EQUAL (ALL IS WELL)
046526  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
046532                                  ERRHRD  ERRNO,T24ILA,EXPREC     ;THE ILA BIT WAS NOT SET IN XST0
046532  104456                                                                  TRAP    C$ERHRD
046534  001063                                                                  .WORD   563
046536  050572                                                                  .WORD   T24ILA
046540  016330                                                                  .WORD   EXPREC
046542                          180$:
046542                                  ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
046542                                                                          L10067:
046542  104403                                                                  TRAP    C$ESUB
046544  023727  002170  000031          CMP     FATFLG,#25.                     ;IS ERROR COUNT AT 25
046552  002402                          BLT     999$                            ;BR, IF LESS THAN 25
046554  004737  020136                  JSR     PC,CKDROP                       ;TRY TO DROP THE UNIT
046560                          999$:
```

```
 8099                                   ;+
 8100                                   ;
 8101                                   ;TEST 5, SUBTEST 10
 8102                                   ;
 8103                                   ;VERIFIES THAT A DATA BUFFER ADDRESS REFERENCING
 8104                                   ;NONEXISTANT MEMORY CAUSES RECOVERABLE ERROR
 8105                                   ;TERMINATION (TC=4), WITH THE NXM BIT SET IN THE TSSR, AND
 8106                                   ;THAT THE TAPE IS ULTIMATELY POSITIONED PROPERLY.
 8107                                   ;
 8108                                   ;
 8109                                   ;
 8110                                   ;-
 8111
 8112 046560                                   BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
      046560                                                                                    T5.10:
      046560  104402                                                                              TRAP    C$BSUB
 8113 046562  012737  000000  050216           MOV     #0,T24DLY               ;SET BITS 16 AND 17
 8114 046570  005737  003100        9$:        TST     KTFLG                   ;CHECK FOR KT11
 8115 046574  001402                           BEQ     10$                     ;BR, IF NO KT11
 8116 046576  000137  047034                   JMP     80$                     ;JUMP IF KT11 (SKIP TEST)
 8117 046602  004737  052664        10$:       JSR     PC,T24RT3               ;SET COMMAND PACKET UP CLEAR
 8118 046606  004737  052530                   JSR     PC,T24REST              ;SET COMMAND PACKET
 8119 046612  004737  052622                   JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET
 8120
 8121                                   ;*************************************************************************
 8122                                   ;
 8123                                   ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
 8124                                   ;
 8125                                   ;*************************************************************************
 8126
 8127 046616  004737  016630                   JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
 8128 046622  103407                           BCS     20$                     ;BR IF INIT WAS OK
 8129 046624  004737  020064                   JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8133 046630  010001                           MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
 8134 046632                                   ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
      046632  104455                                                                            TRAP    C$ERDF
      046634  001064                                                                            .WORD   564
      046636  003550                                                                            .WORD   SFIERR
      046640  011656                                                                            .WORD   SFIMSG
 8135 046642                           20$:
 8136 046642  012704  050040                   MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
 8137
 8138                                   ;*************************************************************************
 8139                                   ;
 8140                                   ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
 8141                                   ;
 8142                                   ;*************************************************************************
 8143
 8144 046646  004737  010322                   JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 8145 046652  103407                           BCS     24$                     ;BR, IF COMMAND ISSUED OK
 8146 046654  004737  020064                   JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8150 046660  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8151 046662                                   ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      046662  104456                                                                            TRAP    C$ERHRD
      046664  001065                                                                            .WORD   565
      046666  004754                                                                            .WORD   WRTMSG
      046670  011656                                                                            .WORD   SFIMSG
```

```
 8152 046672                        24$:    CKLOOP                          ;LOOP IF SELECTED
      046672 104406                                                                  TRAP    C$CLP1
 8153 046674 012701 160000                  MOV     #160000,R1              ;START POSSIBLE NXM ADDRESS
 8154 046700 012702 177776                  MOV     #177776,R2              ;END POSSIBLE NXM ADDRESS
 8155 046704 004737 017260                  JSR     PC,xNXM                 ;CALL NXM FINDER ROUTINE
 8156 046710 103402                         BCS     76$                     ;BR IF WE FOUND NXM
 8157 046712 000137 047034                  JMP     180$                    ;JUMP OVER, CAN'T FIND NXM
 8158 046716 010137 050172          76$:    MOV     R1,T24RB                ;STARTING READ BUFFER ADDRESS
 8159                               ;*************************************************************************
 8160                               ;
 8161                               ;READ, ACK, CVC=1, COMMAND
 8162                               ;
 8163                               ;*************************************************************************
 8164
 8165 046722 013737 050216 050174           MOV     T24DLY,T24RB+2          ;MEMORY BITS 16 AND 17
 8166 046730 012737 140001 050170           MOV     #140001,T24PK3          ;READ, ACK, CVC=1, COMMAND
 8167 046736 012704 050170                  MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
 8168 046742 012737 000400 050176           MOV     #256.,T24SZ             ;SET UP RECORD SIZE IN PACKET
 8169 046750 010465 177776                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
 8170 046754 004737 017104                  JSR     PC,WAITF                ;WAIT FOR SSR!BIT1!BIT2 TO SET
 8171 046760 016501 000000                  MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
 8172 046764 012702 104210                  MOV     #SSR!SC!NXM!BIT3,R2     ;SET UP EXPECTED
 8173 046770 020102                         CMP     R1,R2                   ;ARE THEY EQUAL
 8174 046772 001417                         BEQ     75$                     ;BR, IF OK
 8175 046774 062737 000001 050216           ADD     #1,T24DLY               ;NEXT BUNCH OF MEMORY
 8176 047002 022737 000004 050216           CMP     #4,T24DLY               ;TOO MUCH MEMORY
 8177 047010 001402                         BEQ     50$                     ;END OF LINE
 8178 047012 000137 046602                  JMP     10$                     ;TRY AGAIN
 8179 047016 004737 020064          50$:    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8183 047022                                ERRHRD  ERRNO,T24NXM,PKTSSR     ;TSSR INCORRECT AFTER READ DATA
      047022 104456                                                                 TRAP    C$ERHRD
      047024 001066                                                                 .WORD   566
      047026 050431                                                                 .WORD   T24NXM
      047030 011670                                                                 .WORD   PKTSSR
 8184 047032                        75$:    CKLOOP                          ;LOOP IF SELECTED
      047032 104406                                                                 TRAP    C$CLP1
 8185 047034                        80$:
 8186 047034                        180$:   ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
      047034                                                                        L10070:
      047034 104403                                                                 TRAP    C$ESUB
 8187 047036 023727 002170 000031           CMP     FATFLG,#25.                     ;IS ERROR COUNT AT 25
 8188 047044 002402                         BLT     999$                            ;BR, IF LESS THAN 25
 8189 047046 004737 020136                  JSR     PC,CKDROP                       ;TRY TO DROP THE UNIT
 8190 047052                        999$:
```

```
                                ;+
                                ;
                                ;TEST 5, SUBTEST 11
                                ;
                                ;VERIFIES THAT A READ REVERSE COMMAND ISSUED WHILE THE
                                ;TAPE IS AT BOT RESULTS IN FUNCTION REJECT TERMINATION
                                ;WITH THE NONEXECUTABLE FUNCTION (NEF) ERROR BIT SET
                                ;
                                ;
                                ;
                                ;-

047052                                  BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
047052                                                                                      T5.11:
047052  104402                                                                              TRAP    C$BSUB
047054  004737  052664                    JSR     PC,T24RT3               ;SET COMMAND PACKET UP CLEAR
047060  004737  052530                    JSR     PC,T24REST              ;SET COMMAND PACKET
047064  004737  052622                    JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET

                                ;*************************************************************************
                                ;
                                ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
                                ;
                                ;*************************************************************************

047070  004737  016630                    JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
047074  103407                          BCS     20$                     ;BR IF INIT WAS OK
047076  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
047102  010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
047104                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
047104  104455                                                                              TRAP    C$ERDF
047106  001067                                                                              .WORD   567
047110  003550                                                                              .WORD   SFIERR
047112  011656                                                                              .WORD   SFIMSG
047114                          20$:
047114  012704  050040                    MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS

                                ;*************************************************************************
                                ;
                                ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
                                ;
                                ;*************************************************************************

047120  004737  010322                    JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
047124  103407                          BCS     24$                     ;BR, IF COMMAND ISSUED OK
047126  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
047132  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
047134                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
047134  104456                                                                              TRAP    C$ERHRD
047136  001070                                                                              .WORD   568
047140  004754                                                                              .WORD   WRTMSG
047142  011656                                                                              .WORD   SFIMSG
047144                          24$:    CKLOOP                          ;LOOP IF SELECTED
047144  104406                                                                              TRAP    C$CLP1

                                ;*************************************************************************
                                ;
```

```
 8244                                   ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
 8245                                   ;
 8246                                   ;*************************************************************************
 8247
 8248 047146  004737  010424                    JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
 8249 047152  004737  017220                    JSR     PC,CHKTSSR              ;SEE HOW TSSR IS
 8250 047156  103407                            BCS     30$                     ;BR, IF NO PROBLEM
 8251 047160  010001                            MOV     R0,R1                   ;SAVE TSSR
 8252 047162  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8256 047166                                    ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
      047166  104456                                                                    TRAP    C$ERHRD
      047170  001071                                                                    .WORD   569
      047172  051326                                                                    .WORD   T24RWN
      047174  011670                                                                    .WORD   PKTSSR
 8257 047176                            30$:    CKLOOP                          ;LOOP IF SELECTED
      047176  104406                                                                    TRAP    C$CLP1
 8258 047200  012703  000400                    MOV     #256.,R3                ;RECORD SIZE
 8259 047204  013737  003072  050172            MOV     FREE,T24RB              ;STARTING WRITE BUFFER ADDRESS
 8260
 8261                                   ;*************************************************************************
 8262                                   ;
 8263                                   ;READ REVERSE DATA,ACK COMMAND
 8264                                   ;
 8265                                   ;*************************************************************************
 8266
 8267 047212  012737  100401  050170            MOV     #100401,T24PK3          ;READ REVERSE DATA,ACK COMMAND
 8268 047220  012704  050170                    MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
 8269 047224                            65$:
 8270 047224  010337  050176                    MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
 8271 047230  010465  177776                    MOV     R4,TSDB(R5)             ;ISSUE COMMAND
 8272 047234  004737  017104                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 8273 047240  016501  000000                    MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
 8274 047244  012702  100206                    MOV     #SSR!SC!BIT1!BIT2,R2    ;SET UP EXPECTED
 8275 047250  020102                            CMP     R1,R2                   ;ARE THEY EQUAL
 8276 047252  001406                            BEQ     75$                     ;BR, IF OK
 8277 047254  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8281 047260                                    ERRHRD  ERRNO,T24WDE,PKTSSR     ;TSSR INCORRECT AFTER READ DATA
      047260  104456                                                                    TRAP    C$ERHRD
      047262  001072                                                                    .WORD   570
      047264  050771                                                                    .WORD   T24WDE
      047266  011670                                                                    .WORD   PKTSSR
 8282 047270                            75$:    CKLOOP                          ;LOOP IF SELECTED
      047270  104406                                                                    TRAP    C$CLP1
 8283
 8284                                   ;*************************************************************************
 8285                                   ;
 8286                                   ;READ MESSAGE BUFFER EXTENDED STATUS REGISTER ZERO (XST0)
 8287                                   ;
 8288                                   ;*************************************************************************
 8289
 8290 047272  013701  050066                    MOV     T24BFR+6,R1             ;GET MESSAGE BUFFER
 8291 047276  010102                            MOV     R1,R2                   ;SET UP EXPECTED
 8292 047300  052702  002000                    BIS     #BIT10,R2               ;SET THE NEF BIT IN EXPECTED
 8293 047304  020102                            CMP     R1,R2                   ;ARE THEY EQUAL
 8294 047306  001406                            BEQ     180$                    ;BR, IF EQUAL (ALL IS WELL)
 8295 047310  004737  020064                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8299 047314                                    ERRHRD  ERRNO,T24NEF,EXPREC     ;THE RLL BIT WAS NOT SET IN XST0
```

```
        047314  104456                                                          TRAP    C$ERHRD
        047316  001073                                                          .WORD   571
        047320  050220                                                          .WORD   T24NEF
        047322  016330                                                          .WORD   EXPREC
  8300  047324                        180$:
  8301  047324                              ENDSUB                  ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
        047324                                                                L10071:
        047324  104403                                                          TRAP    C$ESUB
  8302  047326  023727  002170  000031        CMP     FATFLG,#25.             ;IS ERROR COUNT AT 25
  8303  047334  002402                        BLT     999$                    ;BR, IF LESS THAN 25
  8304  047336  004737  020136              JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
  8305  047342                        999$:
```

```
 8307                                   ;+
 8308                                   ;TEST 5, SUBTEST 12
 8309                                   ;VERIFIES THAT A READ REVERSE COMMAND ISSUED WHILE THE
 8310                                   ;TAPE IS POSITIONED BEFORE THE FIRST RECORD ON TAPE
 8311                                   ;(BUT NOT AT BOT) RESULTS IN TAPE STATUS ALERT.
 8312                                   ;-
 8313 047342                            BGNSUB                          ;>>>>>>>>>>>> BEGIN SUBTEST >>>>>>>>>>>>
      047342                                                                    T5.12:
      047342  104402                                                            TRAP    C$BSUB
 8314 047344  004737  052664            JSR     PC,T24RT3               ;SET COMMAND PACKET UP CLEAR
 8315 047350  004737  052530            JSR     PC,T24REST              ;SET COMMAND PACKET
 8316 047354  004737  052622            JSR     PC,T24RT2               ;SET UP OTHER COMMAND PACKET
 8317                                   ;*************************************************************************
 8318                                   ;
 8319                                   ;ISSUE CONTROLLER "SOFT" INITIALIZE - CARRY BIT CLEAR IF ERROR
 8320                                   ;
 8321                                   ;*************************************************************************
 8322 047360  004737  016630            JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
 8323 047364  103407                    BCS     20$                     ;BR IF INIT WAS OK
 8324 047366  004737  020064            JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8328 047372  010001                    MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
 8329 047374                            ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
      047374  104455                                                            TRAP    C$ERDF
      047376  001074                                                            .WORD   572
      047400  003550                                                            .WORD   SFIERR
      047402  011656                                                            .WORD   SFIMSG
 8330 047404  012704  050040    20$:    MOV     #T24PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
 8331                                   ;*************************************************************************
 8332                                   ;
 8333                                   ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
 8334                                   ;
 8335                                   ;*************************************************************************
 8336 047410  004737  010322            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 8337 047414  103407                    BCS     24$                     ;BR, IF COMMAND ISSUED OK
 8338 047416  004737  020064            JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8342 047422  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8343 047424                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      047424  104456                                                            TRAP    C$ERHRD
      047426  001075                                                            .WORD   573
      047430  004754                                                            .WORD   WRTMSG
      047432  011656                                                            .WORD   SFIMSG
 8344 047434                    24$:    CKLOOP                          ;LOOP IF SELECTED
      047434  104406                                                            TRAP    C$CLP1
 8345                                   ;*************************************************************************
 8346                                   ;
 8347                                   ;ISSUE REWIND COMMAND TO SELECTED TAPE DRIVE
 8348                                   ;
 8349                                   ;*************************************************************************
 8350 047436  004737  010424            JSR     PC,REWIND               ;CALL TAPE REWIND COMMAND
 8351 047442  004737  017220            JSR     PC,CHKTSSR              ;SEE HOW TSSR IS
 8352 047446  103407                    BCS     30$                     ;BR, IF NO PROBLEM
 8353 047450  010001                    MOV     R0,R1                   ;SAVE TSSR
 8354 047452  004737  020064            JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8358 047456                            ERRHRD  ERRNO,T24RWN,PKTSSR     ;REWIND NOT ACCEPTED
      047456  104456                                                            TRAP    C$ERHRD
      047460  001076                                                            .WORD   574
      047462  051326                                                            .WORD   T24RWN
```

```
                047464  011670                                                          .WORD   PKTSSR
 8359           047466                  30$:    CKLOOP                          ;LOOP IF SELECTED
                047466  104406                                                          TRAP    C$CLP1
 8360           047470  012703  000400          MOV     #256.,R3                ;RECORD SIZE
 8361           047474  013737  003072  050172  MOV     FREE,T24RB              ;STARTING WRITE BUFFER ADDRESS
 8362                                   ;*************************************************************************
 8363                                   ;
 8364                                   ;WRITE DATA,ACK,CVC=1 COMMAND
 8365                                   ;
 8366                                   ;*************************************************************************
 8367           047502  012737  140005  050170  MOV     #140005,T24PK3          ;WRITE DATA,ACK,CVC=1 COMMAND
 8368           047510  012704  050170          MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
 8369           047514  010337  050176  65$:    MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
 8370           047520  010465  177776          MOV     R4,TSDB(R5)             ;ISSUE COMMAND
 8371           047524  004737  017104          JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 8372           047530  016501  000000          MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
 8373           047534  012702  000200          MOV     #SSR,R2                 ;SET UP EXPECTED
 8374           047540  020102                  CMP     R1,R2                   ;ARE THEY EQUAL
 8375           047542  001406                  BEQ     75$                     ;BR, IF OK
 8376           047544  004737  020064          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8380                                                                           ;SOFT ERROR, REALLY CHECKING THE
 8381                                                                           ;READ REVERSE DATA COMMAND
 8382           047550                          ERRSOFT ERRNO,WRTERR,PKTSSR     ;TSSR INCORRECT AFTER READ DATA
                047550  104457                                                          TRAP    C$ERSOFT
                047552  001077                                                          .WORD   575
                047554  005011                                                          .WORD   WRTERR
                047556  011670                                                          .WORD   PKTSSR
 8383           047560                  75$:    CKLOOP                          ;LOOP IF SELECTED
                047560  104406                                                          TRAP    C$CLP1
 8384           047562  012703  000400          MOV     #256.,R3                ;RECORD SIZE
 8385           047566  013737  003072  050172  MOV     FREE,T24RB              ;STARTING READ BUFFER ADDRESS
 8386                                   ;*************************************************************************
 8387                                   ;
 8388                                   ;READ REVERSE DATA,ACK COMMAND
 8389                                   ;
 8390                                   ;*************************************************************************
 8391           047574  012737  100401  050170  MOV     #100401,T24PK3          ;READ REVERSE DATA,ACK COMMAND
 8392           047602  012704  050170  165$:   MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
 8393           047606  010337  050176          MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
 8394           047612  010465  177776          MOV     R4,TSDB(R5)             ;ISSUE COMMAND
 8395           047616  004737  017104          JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 8396           047622  016501  000000          MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
 8397           047626  012702  000200          MOV     #SSR,R2                 ;SET UP EXPECTED
 8398           047632  020102                  CMP     R1,R2                   ;ARE THEY EQUAL
 8399           047634  001406                  BEQ     170$                    ;BR, IF OK
 8400           047636  004737  020064          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 8404           047642                          ERRHRD  ERRNO,T24TRL,PKTSSR     ;TSSR INCORRECT AFTER READ DATA
                047642  104456                                                          TRAP    C$ERHRD
                047644  001100                                                          .WORD   576
                047646  052374                                                          .WORD   T24TRL
                047650  011670                                                          .WORD   PKTSSR
 8405           047652                  170$:   CKLOOP                          ;LOOP IF SELECTED
                047652  104406                                                          TRAP    C$CLP1
 8406           047654  012703  000400          MOV     #256.,R3                ;RECORD SIZE
 8407           047660  013737  003072  050172  MOV     FREE,T24RB              ;STARTING READ BUFFER ADDRESS
 8408                                   ;*************************************************************************
 8409                                   ;
```

```
8410                                    ;READ REVERSE DATA,ACK COMMAND
8411                                    ;
8412                                    ;*************************************************************************
8413 047666  012737  100401  050170          MOV     #^100401,T24PK3         ;READ REVERSE DATA,ACK COMMAND
8414 047674  012704  050170          195$:   MOV     #T24PK3,R4              ;SET UP R4 WITH PACKET ADDRESS
8415 047700  010337  050176                  MOV     R3,T24SZ                ;SET UP RECORD SIZE IN PACKET
8416 047704  010465  177776                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
8417 047710  004737  017104                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
8418 047714  016501  000000                  MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
8419 047720  012702  100204                  MOV     #SSR!SC!BIT2,R2                 ;SET UP EXPECTED
8420 047724  020102                          CMP     R1,R2                   ;ARE THEY EQUAL
8421 047726  001406                          BEQ     200$                    ;BR, IF OK
8422 047730  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
8426 047734                                  ERRHRD  ERRNO,T24TRL,PKTSSR     ;TSSR INCORRECT AFTER READ DATA
     047734  104456                                                                  TRAP    C$ERHRD
     047736  001101                                                                  .WORD   577
     047740  052374                                                                  .WORD   T24TRL
     047742  011670                                                                  .WORD   PKTSSR
8427 047744                          200$:   CKLOOP                          ;LOOP IF SELECTED
     047744  104406                                                                  TRAP    C$CLP1
8428 047746  013701  050074                  MOV     T24BFR+14,R1            ;GET MESSAGE BUFFER (XST3)
8429 047752  010102                          MOV     R1,R2                   ;SET UP EXPECTED
8430 047754  052702  000001                  BIS     #BIT0,R2                ;SET THE RIB BIT IN EXPECTED
8431 047760  020102                          CMP     R1,R2                   ;ARE THEY EQUAL
8432 047762  001406                          BEQ     210$                    ;BR, IF EQUAL (ALL IS WELL)
8433 047764  004737  020064                  JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
8437 047770                                  ERRHRD  ERRNO,T24LOR,EXPREC     ;THE RIB BIT WAS NOT SET IN XST0
     047770  104456                                                                  TRAP    C$ERHRD
     047772  001102                                                                  .WORD   578
     047774  050272                                                                  .WORD   T24LOR
     047776  016330                                                                  .WORD   EXPREC
8438 050000                          210$:   ENDSUB                          ;>>>>>>>>>>>> END SUBTEST >>>>>>>>>>>>
     050000                                                                         L10072:
     050000  104403                                                                  TRAP    C$ESUB
8439 050002  023727  002170  000031                  CMP     FATFLG,#25.             ;IS ERROR COUNT AT 25
8440 050010  002402                                  BLT     999$                    ;BR, IF LESS THAN 25
8441 050012  004737  020136                          JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
8442 050016                          999$:
8443 050016  004737  017340                  JSR     PC,TSTLOOP              ;DO WE NEED TO ITERATE TEST
8444 050022  103002                          BCC     163$                    ;BR, IF NO LOOP REQUIRED
8445 050024  000137  041514                  JMP     T24LOOP                 ;EXECUTE AGAIN
8446 050030                          163$:
8447 050030                                  EXIT    TST                     ;ALL DONE THIS TEST
     050030  104432                                                                  TRAP    C$EXIT
     050032  002662                                                                  .WORD   L10056-.
```

```
                                        ;+
                                        ;LOCAL STORAGE FOR THIS TEST
                                        ;-
                                                .BLKB   10-<.-TUV2A&7>
                                        T24PACKET:
                                                .WORD   100204          ;COMMAND PACKET FOR TEST
                                                .WORD   T24DATA         ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
                                                .WORD   0               ;ADDRESS OF CHARACTERISTICS BLOCK
                                                .WORD   10.             ;STARTING VALUE OF BLOCK SIZE
                                        T24DATA:                        ;CHARACTERISTICS DATA BLOCK
                                                .WORD   T24BFR          ;ADDRESS OF MESSAGE BUFFER
                                                .WORD   0
                                                .WORD   20.             ;LENGTH OF MESSAGE BUFFER
                                                .WORD   0
                                        T24BFR: .BLKW   25.             ;MESSAGE BUFFER
                                        ;
                                        ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
                                        ;
                                                .BLKB   10-<.-TUV2A&7>
                                        T24PK2:
                                                .WORD   100206          ;WRITE SUB SYS MEM COMMAND, IE AND ACK
                                                .WORD   T24BF2          ;ADDRESS OF SELECT BLOCK DATA
                                                .WORD   0
                                                .WORD   6.              ;SIZE OF DATA PACKET

                                                .BLKB   10-<.-TUV2A&7>
                                        T24PK3:
                                                .WORD   100205          ;READ COMMAND, IE AND ACK
                                        T24RB:
                                        T24WB:  .WORD   FREE            ;ADDRESS OF WRITE BUFFER
                                                .WORD   0
                                        T24SZ:  .WORD   0               ;SIZE OF BUFFER (EXTENT)
                                                .EVEN
                                        ;
                                        ;
                                        ;
                                        T24BF2:
                                        T24BS0: .BYTE   10              ;BSEL0 AREA
                                        T24BS1: .BYTE   200             ;BSEL1 AREA
                                        T24S2:  .WORD   0               ;SEL 2 AREA
                                        T24S3:  .WORD   0               ;DATA AREA
                                        ;
                                        ;
                                                .EVEN
                                        ;TAPE MOTION PACKET COMMAND VALUES

                                        T24RN:  .WORD   100005          ;READ DATA (NEXT)
                                        T24WDR: .WORD   100405          ;READ DATA RETRY
                                        T24CON: .WORD   102005          ;WRITE CONTINOUS
                                                .WORD   177777          ;END OF DATA
                                        T24DLY: .WORD   0               ;DELAY STORAGE AREA
```

```
8509
8510                                    ;+
8511                                    ;LOCAL TEXT MESSAGES FOR TEST
8512                                    ;-
8513
8514 050220    116     105     106     T24NEF: .ASCIZ  'NEF Not Set After NON-EXECUTABLE FUNCTION'
8515 050272    122     111     102     T24LOR: .ASCIZ  'RIB Not Set After READ REVERSE Into BOT'
8516 050342    124     123     123     T24WDG: .ASCIZ  'TSSR Not Correct After Illegal Buffer Address Bits Set'
8517 050431    124     123     123     T24NXM: .ASCIZ  'TSSR Not Correct After NXM Memory Address In Packet'
8518 050515    124     123     123     T24WDF: .ASCIZ  'TSSR Not Correct After Illegal Mode Bits Set'
8519 050572    111     154     154     T24ILA: .ASCIZ  'Illegal Address Bits, Failed To Set ILA Bit In XSTO'
8520 050656    111     154     154     T24LOQ: .ASCIZ  'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
8521 050737    122     105     101     T24SSR: .ASCIZ  'READ COMMAND Not Accepted'
8522 050771    124     123     123     T24WDE: .ASCIZ  'TSSR Not Correct After WRITE DATA Command'
8523 051043    124     141     160     T24BOT: .ASCIZ  'Tape Not At BOT After REWIND Command'
8524 051110    104     141     164     T24DTA: .ASCIZ  'Data Written To Tape Not Equal To Data Read From Tape'
8525 051176    122     105     101     T24EOT: .ASCIZ  'READ DATA OVER EOT GAVE NO TAPE STATUS ALERT'
8526 051253    124     123     123     T24TM:  .ASCIZ  'TSSR Not Correct After READ COMMAND Reject'
8527 051326    122     145     167     T24RWN: .ASCIZ  'Rewind (POSITION) Command Not Accepted'
8528 051375    122     101     115     T24RNC: .ASCIZ  'RAM Error, Correct Data Pattern Not In Ram'
8529 051450    124     123     123     T24AM3: .ASCIZ  'TSSR Init. Failed After READ COMMAND'
8530 051515    104     162     151     T24OFL: .ASCIZ  'Drive 7 Select Failed To Set "OFL" In TSSR'
8531 051570    124     123     123     T24WDD: .ASCIZ  'TSSR Not Correct After READ DATA Command, SWB Bit Set'
8532 051656    124     123     123     T24WDC: .ASCIZ  'TSSR Not Correct After READ DATA Command'
8533 051727    103     126     103     T24VCK: .ASCIZ  'CVC Set, Didn''t Reset VCK In Message Buffer'
8534 052002    124     123     102     T24BA:  .ASCIZ  'TSBA Not Correct After READ DATA Command'
8535 052053    127     122     111     T24WSS: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
8536 052142    122     145     141     T24LON: .ASCIZ  'Reading Long Record Failed To Set RLL Bit In XSTO'
8537 052224    122     145     141     T24LOP: .ASCIZ  'Reading Long Record Failed To Set RLS Bit In XSTO'
8538 052306    122     145     163     T24PBP: .ASCIZ  'Residual Byte Count Incorrect After Short Record Read'
8539 052374    122     145     141     T24TRL: .ASCIZ  'Reading Long Record  Failed To Give Tape Status Alert'
8540 052462    102     141     163     TST24ID:        .ASCIZ  'Basic Read Data (Forward and Reverse)'
8541                                            .EVEN
8542                                    ;+
8543                                    ;
8544                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
8545                                    ;WRITE SUBSYSTEM MEMORY COMMAND
8546                                    ;
8547                                    ;-
8548
8549 052530                             T24REST:
8550 052530                                     SAVREG                          ;SAVE THE REGISTERS
8551 052534    012701  050040                     MOV     #T24PACKET,R1           ;START OF THE PACKET
8552 052540    012721  100004                     MOV     #100004,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK,
8553 052544    012721  050050                     MOV     #T24DATA,(R1)+          ;ADDRESS OF CHARAISTICS DATA BLOCK
8554 052550    005021                             CLR     (R1)+                   ;EXTENDED ADDRESS
8555 052552    012721  000012                     MOV     #10.,(R1)+              ;SIZE OF DATA BLOCK IN BYTES
8556 052556    012721  050060                     MOV     #T24BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
8557 052562    005021                             CLR     (R1)+
8558 052564    012721  000024                     MOV     #20.,(R1)+              ;LENGTH OF MESSAGE BUFFER
8559 052570    005021                             CLR     (R1)+
8560 052572    012711  000000                     MOV     #0,(R1)                 ;SELECT DRIVE ZERO
8561 052576    012702  000030                     MOV     #24.,R2                 ;NUMBER OF LOCATIONS TO BE CLEARED
8562 052602    012762  177777  050060  64$:       MOV     #177777,T24BFR(R2)      ;ALL ONES TO MESSAGE BUFFER
8563 052610    005742                             TST     -(R2)                   ;NEXT LOCATION
8564 052612    022702  000000                     CMP     #0,R2                   ;CKECK FOR END OF LOOP
8565 052616    001371                             BNE     64$                     ;KEEP GOING UNTIL DONE
```

```
 8566 052620  000207                        RTS     PC                      ;RETURN
 8567
 8568
 8569 052622                          T24RT2:
 8570 052622                                  SAVREG                          ;SAVE THE REGISTERS
 8571 052626  012701  050150                  MOV     #T24PK2,R1              ;START OF THE PACKET
 8572 052632  012721  100206                  MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
 8573 052636  012721  050200                  MOV     #T24BF2,(R1)+           ;ADDRESS OF DATA BLOCK
 8574 052642  005021                          CLR     (R1)+                   ;EXTENDED ADDRESS
 8575 052644  012721  000006                  MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
 8576 052650  005021                          CLR     (R1)+
 8577 052652  012701  050200                  MOV     #T24BF2,R1              ;POINT TO DATA SEL AREA
 8578 052656  005021                          CLR     (R1)+
 8579 052660  005011                          CLR     (R1)
 8580 052662  000207                          RTS     PC                      ;RETURN
 8581 052664                          T24RT3:
 8582 052664                                  SAVREG                          ;SAVE THE REGISTERS
 8583 052670  012701  050170                  MOV     #T24PK3,R1              ;START OF THE PACKET
 8584 052674  012721  000000                  MOV     #0,(R1)+                ;CLEAR AREA OUT
 8585 052700  012721  000000                  MOV     #0,(R1)+                ;ADDRESS OF DATA BLOCK
 8586 052704  005021                          CLR     (R1)+                   ;EXTENDED ADDRESS
 8587 052706  012711  000000                  MOV     #0,(R1)                 ;SIZE OF DATA BLOCK IN BYTES
 8588 052712  000207                          RTS     PC                      ;RETURN
 8589 052714                                  ENDTST
      052714
      052714  104401                                              L10056:
                                                                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST 6: MANUAL INTERVENTION

                                ;THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")
                                ;THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
                                ;THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
                                ;THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
                                ;SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
                                ;THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
                                ;ARE BY TYPING <CTRL-C> OR SELECTING CODE 3.
                                ;SELECTION CODES AND SUBROUTINES ARE:
                                ;
                                ;
                                ;       CODE    ROUTINE
                                ;
                                ;        0      HELP. PRINTS THIS MENU.
                                ;        1      REWIND AND UNLOAD COMMAND TEST
                                ;        2      WRITE-PROTECT TEST
                                ;        3      EXIT (RETURN TO SUPERVISOR)
                                ;
                                ;
                                ;EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:
                                ;
                                ;
                                ;
                                ;PRINTS OUT THE MENU ON THE CONSOLE TERMINAL.
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;THIS ROUTINE INSTRUCTS THE OPERATOR TO PLACE THE DRIVE ON-LINE
                                ;AND AT OR BEYOND BOT. THE TEST WILL THEN ISSUE THE REWIND AND
                                ;UNLOAD COMMAND. IT WILL ALSO TELL THE OPERATOR IS THE DRIVE
                                ;ENDED UP ON-LINE OR OFF-LINE.
                                ;
                                ;
                                ;THIS ROUTINE INSTRUCTS THE OPERATOR TO MOUNT A SCRATCH
                                ;TAPE REEL THAT DOES NOT HAVE A WRITE-ENABLE RING INSTALLED, THEN
                                ;WAITS FOR THE OPERATOR TO RESPOND THAT THIS HAS BEEN ACCOMPLISHED.
                                ;UPON THE RESPONSE, THE PROGRAM VERIFIES THAT THE TRANSPORT SHOWS
                                ;A WRITE-PROTECTED STATUS, THEN ATTEMPTS TO WRITE DATA ON THE
                                ;TAPE AND EXPECTS THE APPROPRIATE ERROR TERMINATION INDICATING THAT
                                ;THE WRITE FUNCTION COULD NOT BE PERFORMED BECAUSE THE REEL IS
                                ;WRITE-PROTECTED.  IF THE APPROPRIATE TERMINATION IS NOT RECEIVED,
                                ;AN ERROR IS REPORTED.
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
052716                                  BGNTST
052716                                                                          T6::
052716  005037  002170                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
052722  005037  003100                  CLR     KTFLG           ;HOLD OF KT11
052726  005737  002162                  TST     TSTCNT                  ;IS THIS THE FIRST TEST
```

```
                                                BEQ     21$             ;BR, IF FIRST TEST
                                                MOV     #T38NE,R0       ;"TEST NOT EXECUTED"
                                                BR      3$              ;JUMP IF NOT FIRST TEST
                                        21$:
                                                MOV     #T38ID,R0       ;TEST ID MESSAGE
                                        3$:     JSR     PC,TSTSETUP     ;DO THE COMMON SETUP
                                                JSR     PC,CHKMAN       ;IS MANUAL INTERVENTION ALLOWED?
                                                BCS     19$             ;BR, IF MANUAL INTER ALLOWED
                                                JMP     64$             ;JUMP IF NOT ALLOWED
                                        19$:    CMP     #1,TSTCNT       ;CHECK MIGHT HAVE TO LEAVE
                                                BEQ     22$             ;BR, IF YOU DON'T HAVE TO
                                                JMP     64$             ;WASN'T FIRST TEST IN SEQUENCE
                                        22$:
                                        2$:     CLR     FATFLG          ;CLEAR THE FATAL ERROR FLAG
                                                MOV     #65000.,T38DLY  ;SET UP DELAY COUNTER
                                        5$:     JSR     PC,SOFINIT      ;DO A SOFT INIT
                                                BCS     23$             ;BRANCH IF OK
                                                MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
                                                BIT     #SSR,R1         ;CHECK FOR TSSR SET
                                                BNE     23$             ;KEEP GOING IF NOT SET
                                                DELAY   250             ;CALL DELAY ROUTINE
                                                                                MOV     #250,(PC)+
                                                                                .WORD   0
                                                                                MOV     L$DLY,(PC)+
                                                                                .WORD   0
                                                                                DEC     -6(PC)
                                                                                BNE     .-4
                                                                                DEC     -22(PC)
                                                                                BNE     .-20
                                                DEC     T38DLY          ;BUMP COUNTER DOWN
                                                BNE     5$              ;BR, IF MORE TIME LEFT
                                                ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
                                                                                TRAP    C$ERDF
                                                                                .WORD   601
                                                                                .WORD   SFIERR
                                                                                .WORD   SFIMSG
                                        23$:    MOV     #MIMENU,R0      ;MENU OF MANUAL INTERVENTIONS
                                                MOV     #6,R1           ;MAXIMUM ALLOWED SELECTION
                                                JSR     PC,GETSEL       ;GO GET THE OPERATORS SELECTION
                                                MOV     R0,R4           ;GET NUMBER FROM ROUTINE
                                                ASL     R4              ;CONVERT TO WORD OFFSET
                                                JMP     @6$(R4)         ;JUMP TO PROPER LOOP
                                        6$:     .WORD   2$              ;RETYPE THE MENU
                                                .WORD   20$             ; 3 REWIND AND UNLOAD COMMAND TEST
                                                .WORD   25$             ; 4 WRITE PROTECT
                                                .WORD   63$             ; 6 LEAVE THE TEST


                                        20$:    PRINTF  #T38MS4         ;TELL'EM WHAT TO DO
                                                                                MOV     #T38MS4,-(SP)
                                                                                MOV     #1,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTF
                                                                                ADD     #4,SP
                                        222$:   JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
                                                BCS     300$            ;BR IF SOFT INIT = OK
                                                MOV     R0,R1           ;SAVE CONTENTS OF TSSR
```

```
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
053162  104455                                                                  TRAP    C$ERDF
053164  001132                                                                  .WORD   602
053166  003550                                                                  .WORD   SFIERR
053170  011656                                                                  .WORD   SFIMSG
                                300$:
053172  012704  054470                  MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
053176  004737  010322                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
053202  103405                          BCS     310$                    ;BR, IF COMMAND ISSUED OK
053204  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
053206                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
053206  104456                                                                  TRAP    C$ERHRD
053210  001133                                                                  .WORD   603
053212  004754                                                                  .WORD   WRTMSG
053214  011656                                                                  .WORD   SFIMSG
                                310$:
053216  012737  000144  053756          MOV     #100.,T38DLY            ;SET UP FOR A LONG WAIT
053224  036527  000000  000100  311$:   BIT     TSSR(R5),#OFL           ;IS DRIVE OFF-LINE
053232  001431                          BEQ     315$                    ;BR, IF DRIVE IS ON-LINE
                                        DELAY   250                     ;DELAY ABOUT .25 SEC
053234  012727  000250                                                          MOV     #250,(PC)+
053240  000000                                                                  .WORD   0
053242  013727  002116                                                          MOV     L$DLY,(PC)+
053246  000000                                                                  .WORD   0
053250  005367  177772                                                          DEC     -6(PC)
053254  001375                                                                  BNE     .-4
053256  005367  177756                                                          DEC     -22(PC)
053262  001367                                                                  BNE     .-20
053264  005337  053756                  DEC     T38DLY                  ;BUMP LONG DELAY COUNTER DOWN
053270  001355                          BNE     311$                    ;BR, IF MORE LONG DELAY TO GO
                                        PRINTF  #T38OFL                 ;"DRIVE IS NOW OFF-LINE"
053272  012746  055110                                                          MOV     #T38OFL,-(SP)
053276  012746  000001                                                          MOV     #1,-(SP)
053302  010600                                                                  MOV     SP,R0
053304  104417                                                                  TRAP    C$PNTF
053306  062706  000004                                                          ADD     #4,SP
053312  000137  053152                  JMP     222$                    ;STAY HERE FOREVER, WITH MESSAGE
                                315$:   PRINTF  #T38MS5                 ;"DRIVE SHOULD NOW REWIND AND GO OFL"
053316  012746  055223                                                          MOV     #T38MS5,-(SP)
053322  012746  000001                                                          MOV     #1,-(SP)
053326  010600                                                                  MOV     SP,R0
053330  104417                                                                  TRAP    C$PNTF
053332  062706  000004                                                          ADD     #4,SP
053336  012704  054520                  MOV     #T38PK3,R4              ;SET UP NEW PACKET FOR REWIND/RELEASE
053342  010465  177776                  MOV     R4,TSDB(R5)             ;REWIND RELEASE,ACK,CVC=1 CMD
053346  004737  017104                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
053352  016501  000000                  MOV     TSSR(R5),R1             ;GET TSSR STATUS
053356  012702  000300          320$:   MOV     #SSR!OFL,R2             ;SET UP EXPECTED
053362  020201                          CMP     R2,R1                   ;IS EVERYTHING OK
053364  001404                          BEQ     350$                    ;BR, IF ALL IS WELL
                                        ERRHRD  ERRNO,T38SST,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
053366  104456                                                                  TRAP    C$ERHRD
053370  001134                                                                  .WORD   604
053372  055300                                                                  .WORD   T38SST
053374  011670                                                                  .WORD   PKTSSR
                                350$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
053376  104406                                                                  TRAP    C$CLP1
```

```
                                        MOV     TSSR(R5),R1             ;READ TSSR STATUS
                                        BIT     #OFL,R1                 ;CHECK THE OFF-LINE BIT
                                        BNE     380$                    ;BR, IF DRIVE IS OFF-LINE
                                        PRINTF  #T380NL                 ;"DRIVE IS NOW ON-LINE"
053412  012746  055046                                                          MOV     #T380NL,-(SP)
053416  012746  000001                                                          MOV     #1,-(SP)
053422  010600                                                                  MOV     SP,R0
053424  104417                                                                  TRAP    C$PNTF
053426  062706  000004                                                          ADD     #4,SP
                                        BR      390$                    ;ALMOST DONE
                                380$:   PRINTF  #T380FL                 ;"DRIVE IS NOW OFF-LINE"
053434  012746  055110                                                          MOV     #T380FL,-(SP)
053440  012746  000001                                                          MOV     #1,-(SP)
053444  010600                                                                  MOV     SP,R0
053446  104417                                                                  TRAP    C$PNTF
053450  062706  000004                                                          ADD     #4,SP
                                390$:   CLR     INTRECV                 ;CLEAR INTERRUPT FLAG
                                        JMP     2$                      ;TRY AGAIN
                                25$:    GMANIL  T38MSG,T38DAT,-1,NO     ;WAIT FOR OPERATOR TO MOUNT TAPE
053464  104443                                                                  TRAP    C$GMAN
053466  000404                                                                  BR      10000$
053470  056024                                                                  .WORD   T38DAT
053472  000120                                                                  .WORD   T$CODE
053474  055430                                                                  .WORD   T38MSG
053476  177777                                                                  .WORD   -1
053500                                                                  10000$:
                                        BNCOMPLETE 25$                  ;RETRY IF ERROR
053500  103371                                                                  BCC     25$
                                        TST     T38DAT                  ;DID OPERATOR SAY 'YES' ?
                                        BNE     27$                     ;BRANCH IF YES
                                        JMP     2$                      ;RETURN TO MAIN MENU
                                27$:
                                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
                                        BCS     400$                    ;BR IF SOFT INIT = OK
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
053524  104455                                                                  TRAP    C$ERDF
053526  001135                                                                  .WORD   605
053530  003550                                                                  .WORD   SFIERR
053532  011656                                                                  .WORD   SFIMSG
                                400$:   CKLOOP                          ;LOOP IF SELECTED
053534  104406                                                                  TRAP    C$CLP1
                                        MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
                                        JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
                                        BCS     410$                    ;BR, IF COMMAND ISSUED OK
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
053552  104456                                                                  TRAP    C$ERHRD
053554  001136                                                                  .WORD   606
053556  004754                                                                  .WORD   WRTMSG
053560  011656                                                                  .WORD   SFIMSG
                                410$:   CKLOOP                          ;LOOP IF SELECTED
053562  104406                                                                  TRAP    C$CLP1
                                        MOV     T38BFR+6,R1             ;PICK UP XSTO CONTENTS
                                        MOV     R1,R2                   ;SET UP EXPECTED
                                        BIS     #BIT2,R2                ;SET UP THE WRITE LOCKED BIT
                                        CMP     R1,R2                   ;ARE THEY CORRECT
```

```
 8764 053600  001406                        BEQ     430$                    ;BR, IF ALL IS WELL (OK)
 8768 053602                                 ERRHRD  ERRNO,T38WRL,EXPREC     ;"WRITE LOCKED BIT IS NOT SET ETC."
      053602  104456                                                                TRAP    C$ERHRD
      053604  001137                                                               .WORD   607
      053606  054711                                                               .WORD   T38WRL
      053610  016330                                                               .WORD   EXPREC
 8769 053612  000137  053000                    JMP     2$                      ;BECAUSE OF ERROR GO BACK TO MENU
 8770 053616                         430$:      CKLOOP                          ;LOOP IF SELECTED
      053616  104406                                                                TRAP    C$CLP1
 8771 053620                                 PRINTF  #T38WOK                 ;PRINT "DRIVE IS WRITE PROTECTED"
      053620  012746  055470                                                        MOV     #T38WOK,-(SP)
      053624  012746  000001                                                        MOV     #1,-(SP)
      053630  010600                                                               MOV     SP,R0
      053632  104417                                                               TRAP    C$PNTF
      053634  062706  000004                                                        ADD     #4,SP
 8772 053640  017737  127226  054542  435$:  MOV     @FREE,T38WR             ;SET UP WRITE BUFFER ADDRESS
 8773 053646  012704  054540                MOV     #T38PK4,R4              ;GET PACKET ADDRESS
 8774 053652  010465  177776                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 8775 053656  004737  017104                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 8776 053662  016501  000000                MOV     TSSR(R5),R1             ;GET TSSR
 8777 053666  012702  100206                MOV     #SC!SSR!BIT1!BIT2,R2    ;SET UP EXPECTED
 8778 053672  020102                        CMP     R1,R2                   ;ARE THEY EQUAL (CORRECT)
 8779 053674  001404                        BEQ     440$                    ;BR, IF CORRECT STATUS
 8783 053676                                ERRHRD  ERRNO,T38WRT,PKTSSR     ;"TSSR INCORRECT AFTER WRITE COMMAND
      053676  104456                                                                TRAP    C$ERHRD
      053700  001140                                                               .WORD   608
      053702  054625                                                               .WORD   T38WRT
      053704  011670                                                               .WORD   PKTSSR
 8784 053706                         440$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      053706  104406                                                                TRAP    C$CLP1
 8785 053710  013701  054014                MOV     T38BFR+6,R1             ;READ XSTO CONTENTS
 8786 053714  010102                        MOV     R1,R2                   ;SET UPR EXPECTED
 8787 053716  052702  004000                BIS     #BIT11,R2               ;SET THE WRITE LOCK ERROR BIT (XSTO)
 8788 053722  020102                        CMP     R1,R2                   ;WAS THE BIT SET
 8789 053724  001404                        BEQ     450$                    ;BR, IF IT WAS (GOOD)
 8793 053726                                ERRHRD  ERRNO,T38WLE,EXPREC     ;"WRITE LOCK ERROR BIT NOT SET"
      053726  104456                                                                TRAP    C$ERHRD
      053730  001141                                                               .WORD   609
      053732  054752                                                               .WORD   T38WLE
      053734  016330                                                               .WORD   EXPREC
 8794 053736                         450$:      CKLOOP                          ;LOOP IF SELECTED
      053736  104406                                                                TRAP    C$CLP1
 8795 053740  000137  053000                    JMP     2$                      ;GO BACK TO MENU
 8796
 8797
 8798
 8799 053744                         63$:
 8800 053744                         64$:       EXIT    TST                     ;LEAVE TEST
      053744  104432                                                                TRAP    C$EXIT
      053746  002570                                                               .WORD   L10073-.
 8801
 8802                                ;+
 8803                                ;LOCAL TEXT MESSAGES FOR TEST
 8804                                ;-
 8805
 8806                                ;LOCAL STORAGE FOR THIS TEST
 8807                                ;-
```

```
                                        ;+
                                        ;LOCAL STORAGE FOR THIS TEST
                                        ;-

053750  000000                          TTION2:         .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
053752  000000                          TVSAV2: .WORD   0               ;SAVE TTI VECTOR
053754  000000                          TPSAV2: .WORD   0               ;SAVE TTI PRIORITY

053756  000000                          T38DLY: .WORD   0                       ;DELAY COUNTER FOR TEST
053760                                          .BLKB   10-<.-TUV2A&7>
053770                                  T38PACKET:                              ;COMMAND PACKET FOR TEST
053770  140006                                  .WORD   140006                  ;WRITE SUBSYSTEM MEM. CMD.  ACK,CVC=1
053772  054000                                  .WORD   T38TAD                  ;ADDRESS OF CHARACTERISTICS BLOCK
053774  000000                                  .WORD   0
053776  000012                                  .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
054000                                  T38TAD:                         ;CHARACTERISTICS DATA BLOCK
054000  000                             T38BS0: .BYTE   0                       ;BSEL0 BYTE
054001  000                             T38BS1: .BYTE   0                       ;BSEL1 BYTE
054002  000000                          T38BS2: .WORD   0                       ;BSEL1 WORD
054004  000000                                  .WORD   0                       ;DATA
054006                                  T38BFR: .BLKW   150.                    ;MESSAGE BUFFER
054462  000000                          T38EB:  .WORD                           ;END OF BUFFER ADDRESS


                                                .BLKB   10-<.-TUV2A&7>
054464
054470                                  T38PK2:                                 ;COMMAND PACKET FOR TEST
054470  140004                                  .WORD   140004                  ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
054472  054500                                  .WORD   T38DTA                  ;ADDRESS OF SELECT DATA BLOCK
054474  000000                                  .WORD   0
054476  000012                                  .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE


054500                                  T38DTA:                                 ;SELECT DATA BLOCK
054500  054006                                  .WORD   T38BFR                  ;ADDRESS OF MESSAGE BUFFER
054502  000000                                  .WORD   0
054504  000400                                  .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
054506  000000                          T38EAI: .WORD   0                       ;EAI BIT WORD
054510                                          .BLKB   10-<.-TUV2A&7>
054520  140412                          T38PK3: .WORD   140412                  ;REWIND AND UNLOAD COMMAND
054522  000000                                  .WORD   0                       ;NOT USED
054524  000000                                  .WORD   0                       ;NOT USED
054526  000000                                  .WORD   0                       ;NOT USED
054530  000000                                  .WORD   0                       ;NOT USED
                                        ;
                                        ;WRITE TAPE PACKET
                                        ;
054532                                          .BLKB   10-<.-TUV2A&7>
054540  140005                          T38PK4: .WORD   140005                  ;WRITE, ACK, CVC=1 COMMAND
054542  000000                          T38WR:  .WORD   0                       ;ADDRESS OF WRITE BUFFER
054544  000000                                  .WORD   0                       ;MORE ADDRESS OF WRITE BUFFER
054546  000400                          T38SIZ: .WORD   256.                    ;SIZE OF RECORD


                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
```

```
                                        ;-
                                        T38NE:  .ASCIZ  'Stand-alone Manual Intervention Not Executed'
                                        T38WRT: .ASCIZ  'TSSR Not Correct After WRITE, With WRITE PROTECT On'
                                        T38WRL: .ASCIZ  'WRITE LOCKED Bit Not Set In XSTO'
                                        T38WLE: .ASCIZ  'WRITE LOCK ERROR Bit Not Set In XSTO, After Attempted WRITE'
                                        T38ONL: .ASCIZ  '%N%A ERROR Drive Is Still ON-LINE'
                                        T38OFL: .ASCIZ  '%N%A Drive Is Now  OFF-LINE'
                                        T38MS4: .ASCIZ  '%N%A Set Drive To On-line and At Or Beyond BOT'
                                        T38MS5: .ASCIZ  '%N%A Drive Should Now Rewind and Go Off-line'
                                        T38SST: .ASCIZ  'Contents Of TSSR Incorrect After REWIND And RELEASE'
                                        T38MS2: .ASCIZ  '%N%AType RETURN To Return To Menu%N'
                                        T38MSG: .ASCIZ  'Is Write-Protected Tape Mounted'
                                        T38WOK: .ASCIZ  '%N%A Drive Is Write Protected'
                                        T38ID:  .ASCIZ  'Manual Intervention'
                                                .EVEN
                                        MIMENU: .WORD   1$,2$,5$,6$
                                                .WORD   9$,10$,0
                                        1$:     .ASCIZ  <12>'SELECT OPERATION FROM FOLLOWING OPTIONS:'
                                        2$:     .ASCIZ  <12>'   0       Display This Menu'
                                        5$:     .ASCIZ  '       1       Rewind and Unload Command Test'
                                        6$:     .ASCIZ  '       2       Write Protect Test'
                                        9$:     .ASCIZ  '       3       Return to Diagnostic Supervisor'
                                        10$:    .ASCIZ  ''
                                                .EVEN


                                        ;+
                                        ;LOCAL STORAGE FOR THIS TEST
                                        ;-

                                        T38DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
                                        T38REST:
                                                SAVREG                  ;SAVE THE REGISTERS
                                                MOV     #T38PACKET,R1   ;START OF THE PACKET
                                                MOV     #140206,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
                                                MOV     #T38TAD,(R1)+   ;ADDRESS OF DATA BLOCK
                                                CLR     (R1)+           ;EXTENDED ADDRESS
                                                MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
                                                CLR     (R1)+           ;CLEAR BSEL0 AND BSEL1
                                                CLR     (R1)+           ;CLEAR SEL2
                                                CLR     (R1)            ;CLEAR DATA AREA
                                                RTS     PC              ;RETURN


                                        ;+
                                        ;
                                        ;THIS ROUTINE PRINTS THE CONTENTS OF
                                        ;THE 256 BYTE MESSAGE BUFFER RETURNED BY THE
                                        ;TUV-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
                        ;
                        ;-

                        T38MBP:
                                SAVREG                  ;SAVE THE REGISTERS
                                MOV     R0,R5           ;SAVE LOW ORDER ADDRESS
                                TST     KTENABLE        ;ADDRESS ABOVE 28K?
                                BNE     910$            ;BR IF YES
                                CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
                        910$:   MOV     R1,R3           ;SAVE HIGH ORDER ADDRESS
                                ROL     R0              ;SHIFT BIT15 TO C BIT
                                ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
                                PRINTX  #T38AS0,R1,R5   ;PRINT MESSAGE BUFFER ADDRESS
                                                                        MOV     R5,-(SP)
                                                                        MOV     R1,-(SP)
                                                                        MOV     #T38AS0,-(SP)
                                                                        MOV     #3,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #10,SP
                                PRINTX  #T38AS1         ;PRINT HEADER FOR CONTENTS
                                                                        MOV     #T38AS1,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #4,SP
                                MOV     R5,R1           ;COPY LOW ORDER ADDRESS
                                MOV     R3,R0           ;COPY HIGH ORDER ADDRESS
                                BEQ     913$            ;BR IF NOT ABOVE 28K
                                JSR     PC,SETMAP       ;SETUP PAR ADDRESS IN R0
                                MOV     R0,R5           ;GET PAR FORMAT ADDRESS ABOVE 28K
                        913$:   MOV     R5,T38CNT       ;HOLD ADDRESS
                        911$:   MOV     (R5),R4         ;GET BUFFER ENTRY
                                CMP     #125252,R4      ;CHECK FOR NO LOAD CONDITION
                                BEQ     912$            ;BR, IF BUFFER WASN'T LOADED
                                MOV     R4,R3           ;MAKE COPY
                                BIC     #170377,R4      ;ONLY BITS 11,10,9 AND 8 ARE SAVED
                                CLC                     ;CLEAR CARRY
                                ROR     R4              ;11 TO 10 BIT POSITION
                                ROR     R4              ;10 TO 9 BIT POSITION
                                ROR     R4              ;9 TO 8 BIT POSITON
                                ROR     R4              ;8 TO 7 BIT POSITION
                                BIC     #177760,R3      ;ONLY BITS 3,2,1 AND 0 ARE SAVED
                                ADD     R4,R3           ;'OR'EM TOGETHER
                                MOV     R3,(R5)+        ;PUT BACK IN BUFFER
                                CMP     R5,#T38EB       ;END OF BUFFER YET
                                BNE     911$            ;BR, IF NOT AT END YET
                        912$:   MOV     T38CNT,R5       ;PUT ADDRESS BACK
                                MOV     #1,R4           ;START BYTE NUMBER AT ONE
                        915$:   PRINTX  #T38ASN,R4,(R5)+        ;PRT MEM BUFFER W/NEWLINE
                                                                        MOV     (R5)+,-(SP)
                                                                        MOV     R4,-(SP)
                                                                        MOV     #T38ASN,-(SP)
```

```
     056262  012746  000003                                                              MOV     #3,-(SP)
     056266  010600                                                                      MOV     SP,R0
     056270  104415                                                                      TRAP    C$PNTX
     056272  062706  000010                                                              ADD     #10,SP
     056276  005037  056534                  CLR     T38CNT                   ;CLEAR COUNTER
     056302  000412                          BR      921$                     ;SKIP OTHER PRINT
     056304                          920$:   PRINTX  #T38ASC,R4,(R5)+         ;PRINT THE CONTENTS OF MEMORY BUFFER
     056304  012546                                                                      MOV     (R5)+,-(SP)
     056306  010446                                                                      MOV     R4,-(SP)
     056310  012746  056471                                                              MOV     #T38ASC,-(SP)
     056314  012746  000003                                                              MOV     #3,-(SP)
     056320  010600                                                                      MOV     SP,R0
     056322  104415                                                                      TRAP    C$PNTX
     056324  062706  000010                                                              ADD     #10,SP
     056330  005237  056534          921$:   INC     T38CNT                   ;BUMP COUNTER
     056334  005204                          INC     R4               ;NUMBER OF THE NEXT
     056336  020427  000200                  CMP     R4,#128.                 ;DONE ALL YET ?
     056342  003010                          BGT     50$              ;BRANCH IF ALL DONE
     056344  023727  056534  000004          CMP     T38CNT,#4                ;DONE FOUR YET
     056352  001401                          BEQ     925$                     ;BR, IF THREE DONE
     056354  000753                          BR      920$             ;KEEP GOING
     056356  005037  056534          925$:   CLR     T38CNT                   ;CLEAR COUNTER
     056362  000733                          BR      915$                     ;PRINT WITH NEW LINE
     056364  000207                  50$:    RTS     PC               ;RETURN

     056366     045     116     045  T38AS0: .ASCIZ  '%N%A Message Buffer Address = %O1%O5'
     056433     045     116     045  T38AS1: .ASCIZ  '%N%A Message Buffer Contents:'
     056471     045     101     040  T38ASC: .ASCIZ  '%A  %D4%A: %O3'
     056510     045     116     045  T38ASN: .ASCIZ  '%N%A Byte%D4%A: %O3'
                                             .EVEN
     056534  000000                  T38CNT: .WORD                    ;COUNTER FOR PRINT
     056536                                  ENDTST
     056536                                                                  L10073:
     056536  104401                                                                      TRAP    C$ETST
```

```
                                .SBTTL  TEST 7: CONFIGURATION TYPEOUT

                              ;THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL
                              ;THE CONFIGURATION OF THE M7454 MODULE AND TUV05 SUBSYSTEM. SPECIFICALLY,
                              ;THE FOLLOWING INFORMATION IS PRESENTED:
                              ;
                              ;
                              ; 1.0 MICROCODE REVSION LEVEL OF THE M7454.
                              ;
                              ; 2.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.
                              ;
                              ; 3.0 UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED)
                              ;     OF EACH CONNECTED TRANSPORT.
                              ;
                              ;
                              ;THE OPERATOR IS EXPECTED TO READ THE PRINTOUT AND VERIFY THAT IT MATCHES
                              ;THE ACTUAL CONFIGURATION AT HAND.  IF, FOR EXAMPLE, THE PROGRAM INDICATES
                              ;THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN FACT ONLY ONE IS PRESENT,
                              ;THE OPERATOR MUST INTERPRET THIS AS AN ERROR AND ATTEMPT TO FIND THE
                              ;CAUSE (BAD CABLE, FAULTY UNIT-SELECT DECODING IN THE TRANSPORT, ETC.).
                              ;[SINCE THE CONTROLLER CAN ONLY ACCESS UNIT 0 IF IT IS IN "STANDARD"
                              ;MODE, THE PROGRAM WILL FORCE THE MODULE INTO EXTENDED MODE VIA THE
                              ;WRITE SUBSYSTEM MEMORY COMMAND IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]
                              ;
                              ;
                              ;THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO THE
                              ;DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE
                              ;SELECTED (E.G., FROM THE INITIAL STARTUP DIALOG), THE ROUTINE WILL BE
                              ;REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.
                              ;
056540                              BGNTST                                                  T7::
056540
056540 005037 002170                CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
056544 005037 003100                CLR     KTFLG           ;HOLD OF KT11
056550 005737 002162                TST     TSTCNT                  ;IS THIS FIRST TEST IN SEQUENCE ?
056554 001403                       BEQ     10$                     ;BR, IF FIRST TEST
056556 012700 057773                MOV     #T39NE,R0               ;"TEST NOT EXECUTED"
056562 000402                       BR      11$                     ;JUMP OUT OF TEST IF NOT
056564 012700 060432        10$:    MOV     #TST39ID,R0             ;TEST ID MESSAGE
056570 004737 017372        11$:    JSR     PC,TSTSETUP             ;DO THE COMMON SETUP
056574 004737 021366                JSR     PC,CHKMAN               ;IS MANUAL INTERVENTION ALLOWED?
056600 103402                       BCS     15$                     ;BR, IF MANUAL INTERVENTION ALLOWED
056602 000137 057202                JMP     64$                     ;JUMP TO OUT IF NOT
056606 022737 000001 002162 15$:    CMP     #1,TSTCNT               ;IS THIS THE FIRST TEST IN SEQ
056614 001402                       BEQ     20$                     ;BR, IF FIRST TEST
056616 000137 057202                JMP     64$                     ;JMP IF IT WASN'T
056622                      20$:
056622 004737 016630                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
056626 103405                       BCS     25$                     ;BR IF SOFT INIT = OK
056630 010001                       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
056632                              ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
056632 104455                                                               TRAP    C$ERDF
056634 001275                                                               .WORD   701
056636 003550                                                               .WORD   SFIERR
056640 011656                                                               .WORD   SFIMSG
056642                      25$:    CKLOOP                                  ;LOOP IF SELECTED
```

```
     056642 104406                                                              TRAP    C$CLP1
     056644 012704 057720               MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
     056650 004737 010322               JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
     056654 103405                      BCS     50$                     ;BR, IF COMMAND ISSUED OK
     056656 010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     056660                             ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     056660 104456                                                              TRAP    C$ERHRD
     056662 001276                                                              .WORD   702
     056664 004754                                                              .WORD   WRTMSG
     056666 011656                                                              .WORD   SFIMSG
     056670                     50$:    CKLOOP                          ;LOOP IF SELECTED
     056670 104406                                                              TRAP    C$CLP1
     056672 013701 057242       130$:   MOV     T39BFR+4,R1             ;PICK UP THE RES. BYTE CNTR AREA
     056676 042701 177700               BIC     #177700,R1              ;ONLY LEAVE MICROCODE REV LEVEL
     056702 010137 060370               MOV     R1,T39RL                ;LOAD UP REV LEVEL
     056706                             PRINTX  #T39MCL,T39RL           ;"MICROCODE REVISION LEVEL =000XXX"
     056706 013746 060370                                                       MOV     T39RL,-(SP)
     056712 012746 060313                                                       MOV     #T39MCL,-(SP)
     056716 012746 000002                                                       MOV     #2,-(SP)
     056722 010600                                                              MOV     SP,R0
     056724 104415                                                              TRAP    C$PNTX
     056726 062706 000006                                                       ADD     #6,SP
     056732 004737 016630               JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
     056736 103405                      BCS     140$                    ;BR IF SOFT INIT = OK
     056740 010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     056742                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     056742 104455                                                              TRAP    C$ERDF
     056744 001277                                                              .WORD   703
     056746 003550                                                              .WORD   SFIERR
     056750 011656                                                              .WORD   SFIMSG
     056752                     140$:   CKLOOP                                  ;LOOP IF SELECTED
     056752 104406                                                              TRAP    C$CLP1
     056754 012704 057720               MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
     056760 004737 010322               JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
     056764 103405                      BCS     150$                    ;BR, IF COMMAND ISSUED OK
     056766 010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     056770                             ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     056770 104456                                                              TRAP    C$ERHRD
     056772 001300                                                              .WORD   704
     056774 004754                                                              .WORD   WRTMSG
     056776 011656                                                              .WORD   SFIMSG
     057000                     150$:   CKLOOP                          ;LOOP IF SELECTED
     057000 104406                                                              TRAP    C$CLP1
     057002 005037 002150               CLR     UNITN                   ;SET TO DRIVE 0

     057006 016501 000000       190$:   MOV     TSSR(R5),R1             ;GET TSSR STATUS
     057012 032701 000100               BIT     #OFL,R1                 ;CHECK FOR OFF-LINE
     057016 001414                      BEQ     200$                    ;BR, IF DRIVE IS ON-LINE
     057020                             PRINTX  #T39OF2,UNITN           ;"DRIVE NUMBER XX IS OFF-LINE"
     057020 013746 002150                                                       MOV     UNITN,-(SP)
     057024 012746 060052                                                       MOV     #T39OF2,-(SP)
     057030 012746 000002                                                       MOV     #2,-(SP)
     057034 010600                                                              MOV     SP,R0
     057036 104415                                                              TRAP    C$PNTX
     057040 062706 000006                                                       ADD     #6,SP
     057044 000137 057162               JMP     63$                     ;DO NOT TRY TO GET ANYMORE INFO.
     057050                     200$:   PRINTX  #T39ON2,UNITN           ;"DRIVE NUMBER XX IS ON-LINE"
```

```
                057050  013746  002150                                                  MOV     UNITN,-(SP)
                057054  012746  060116                                                  MOV     #T390N2,-(SP)
                057060  012746  000002                                                  MOV     #2,-(SP)
                057064  010600                                                          MOV     SP,R0
                057066  104415                                                          TRAP    C$PNTX
                057070  062706  000006                                                  ADD     #6,SP
9091            057074  013701  057244          MOV     T39BFR+6,R1         ;READ EXTENDED STATUS (XSTO)
9092            057100  032701  000004          BIT     #BIT2,R1            ;IS DRIVE WTITE PROTECTED
9093            057104  001013                  BNE     210$                ;BR, IF WRITE PROTECTED
9094            057106                          PRINTX  #T39WPN,UNITN       ;"DRIVE NUMBER IS NOT WRT PRO"
                057106  013746  002150                                                  MOV     UNITN,-(SP)
                057112  012746  060234                                                  MOV     #T39WPN,-(SP)
                057116  012746  000002                                                  MOV     #2,-(SP)
                057122  010600                                                          MOV     SP,R0
                057124  104415                                                          TRAP    C$PNTX
                057126  062706  000006                                                  ADD     #6,SP
9095            057132  000413                  BR      63$                 ;SKIP OVER
9096            057134                  210$:   PRINTX  #T39WRT,UNITN       ;"DRIVE NUMBER XX IS WRT PRO"
                057134  013746  002150                                                  MOV     UNITN,-(SP)
                057140  012746  060161                                                  MOV     #T39WRT,-(SP)
                057144  012746  000002                                                  MOV     #2,-(SP)
                057150  010600                                                          MOV     SP,R0
                057152  104415                                                          TRAP    C$PNTX
                057154  062706  000006                                                  ADD     #6,SP
9097            057160  000400                  BR      63$                 ;BR, IF NO MORE DRIVES
9098            057162                  63$:    PRINTX  #T39NFL                     ;NEW LINE
                057162  012746  057770                                                  MOV     #T39NFL,-(SP)
                057166  012746  000001                                                  MOV     #1,-(SP)
                057172  010600                                                          MOV     SP,R0
                057174  104415                                                          TRAP    C$PNTX
                057176  062706  000004                                                  ADD     #4,SP
9099            057202                  64$:    EXIT    TST                 ;EXIT THIS SECTION
                057202  104432                                                          TRAP    C$EXIT
                057204  001254                                                          .WORD   L10074-.
9100                                    ;+
9101                                    ;LOCAL TEXT MESSAGES FOR TEST
9102                                    ;-
9103
9104                                    ;LOCAL STORAGE FOR THIS TEST
9105                                    ;-
9106
9107            057206  000000          T39DLY: .WORD   0                   ;DELAY COUNTER FOR TEST
9109            057210                          .BLKB   10-<.-TUV2A&7>
9111            057220                  T39PACKET:                          ;COMMAND PACKET FOR TEST
9112            057220  140006                  .WORD   140006              ;WRITE SUBSYSTEM MEM. CMD,  ACK,CVC=1
9113            057222  057230                  .WORD   T39TAD              ;ADDRESS OF CHARACTERISTICS BLOCK
9114            057224  000000                  .WORD   0
9115            057226  000012                  .WORD   10.                 ;STARTING VALUE OF BLOCK SIZE
9116            057230                  T39TAD:                     ;CHARACTERISTICS DATA BLOCK
9117            057230     000          T39BS0: .BYTE   0                   ;BSEL0 BYTE
9118            057231     000          T39BS1: .BYTE   0                   ;BSEL1 BYTE
9119            057232  000000          T39BS2: .WORD   0                   ;BSEL1 WORD
9120            057234  000000                  .WORD   0                   ;DATA
9121            057236                  T39BFR: .BLKW   150.                ;MESSAGE BUFFER
9122
9123
9125            057712                          .BLKB   10-<.-TUV2A&7>
```

```
                                T39PK2:                                 ;COMMAND PACKET FOR TEST
                                        .WORD   140004                  ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
                                        .WORD   T39DTA                  ;ADDRESS OF SELECT DATA BLOCK
                                        .WORD   0
                                        .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE


                                T39DTA:                                 ;SELECT DATA BLOCK
                                        .WORD   T39BFR                  ;ADDRESS OF MESSAGE BUFFER
                                        .WORD   0
                                        .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
                                T39EAI: .WORD   0                       ;EAI BIT WORD
                                        .BLKB   10-<.-TUV2A&7>
                                T39PK3: .WORD   140012                  ;MESSAGE BUFFER RELEASE COMMAND
                                        .WORD   0                       ;NOT USED
                                ;
                                ;WRITE TAPE PACKET
                                ;
                                        .BLKB   10-<.-TUV2A&7>
                                T39PK4: .WORD   140005                  ;WRITE, ACK, CVC=1 COMMAND
                                T39WR:  .WORD   0                       ;ADDRESS OF WRITE BUFFER
                                        .WORD   0                       ;MORE ADDRESS OF WRITE BUFFER
                                T39SIZ: .WORD   256.                    ;SIZE OF RECORD




                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-


 057770      045     116     000  T39NFL: .ASCIZ  '%N'
 057773      123     164     141  T39NE:  .ASCIZ  'Stand-alone Configuration Typeout Not Executed'
 060052      045     116     045  T39OF2: .ASCIZ  '%N%A Drive Number %D2%A Is Off-Line'
 060116      045     116     045  T39ON2: .ASCIZ  '%N%A Drive Number %D2%A Is On-Line'
 060161      045     116     045  T39WRT: .ASCIZ  '%N%A Drive Number %D2%A Is Write Protected'
 060234      045     116     045  T39WPN: .ASCIZ  '%N%A Drive Number %D2%A Is NOT Write Protected'
 060313      045     116     045  T39MCL: .ASCIZ  '%N%A M7454 Microcode Revision Level     =%02'
                                        .EVEN
 060370  000000                  T39RL:  .WORD   0
                                        .EVEN
                                        .EVEN


                                ;+
                                ;LOCAL STORAGE FOR THIS TEST
                                ;-

 060372  000000                  T39DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
 060374                          T39REST:
 060374                                  SAVREG                  ;SAVE THE REGISTERS
 060400  012701  057220                  MOV     #T39PACKET,R1   ;START OF THE PACKET
 060404  012721  140006                  MOV     #140006,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
 060410  012721  057230                  MOV     #T39TAD,(R1)+   ;ADDRESS OF DATA BLOCK
 060414  005021                          CLR     (R1)+           ;EXTENDED ADDRESS
```

```
 9188 060416  012721  000006                MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
 9189 060422  005021                        CLR     (R1)+                   ;CLEAR BSEL0 AND BSEL1
 9190 060424  005021                        CLR     (R1)+                   ;CLEAR SEL2
 9191 060426  005011                        CLR     (R1)                    ;CLEAR DATA AREA
 9192 060430  000207                        RTS     PC                      ;RETURN
 9193
 9194                                ;+
 9195                                ;LOCAL TEXT MESSAGES FOR TEST
 9196                                ;-
 9197
 9198 060432     103     157     156 TST39ID:        .ASCIZ  'Configuration Typeout'
 9199                                .EVEN
 9200 060460                        ENDTST
      060460                                                          L10074:
      060460  104401                                                          TRAP    CSETST
 9201
```

```
                                .SBTTL  TEST 8: SCOPE LOOPS


                        ;+
                        ;
                        ;       THIS IS A STANDALONE ROUTINE PROVIDING A NUMBER OF TIGHT "SCOPE
                        ;       LOOPS" USEFUL FOR DEBUGGING BASIC REGISTER ACCESS PROBLEMS WITH
                        ;       THE M7454 MODULE.  THESE SCOPE LOOPS CAN BE USED WHEN THE NORMAL
                        ;       "LOOP ON ERROR" OR "LOOP ON TEST (SUBTEST)" FACILITIES DON'T
                        ;       SEEM TO ALLOW THE OPERATOR TO ZERO IN A PROBLEM IN THE EARLY
                        ;       TESTS (I.E. THE HARDWARE MAY NOT BE RESPONDING TO A REGISTER
                        ;       ACCESS, CAUSING A BUS ERROR TRAP, EVEN THOUGH THE DEVICE ADDRESS
                        ;       SELECTED BY THE PROGRAM MATCHES THE CONFIGURATION SET UP IN THE
                        ;       HARDWARE DIP SWITCHES).  THE FOLLOWING MENU OF SCOPE LOOPS ARE
                        ;       AVAILABLE:
                        ;
                        ;               CODE    SCOPE LOOP
                        ;
                        ;                 0     HELP. DISPLAY THIS MENU.
                        ;                 1     TSBA READ ACCESS
                        ;                 2     TSSR READ ACCESS
                        ;                 3     INITIALIZE (TSSR WRITE ACCESS)
                        ;                 4     TSDB HIGH BYTE WRITE ACCESS
                        ;                 5     TSDB LOW BYTE WRITE ACCESS
                        ;                 6     TSSR BYTE WRITE (SELF-TEST)
                        ;                 7     RETURN TO DIAGNOSTIC SUPERVISOR
                        ;
                        ;       FOR SCOPE LOOPS THAT WRITE INTO REGISTERS, THE PROGRAM PROMPTS
                        ;       THE OPERATOR FOR THE DATA TO BE WRITTEN.  TYPING <RETURN> CAUSES
                        ;       AN EXIT FROM THE SCOPE LOOP BACK TO MENU LEVEL.
                        ;-


060462                  BGNTST                                                  T8::
060462  005037  002170          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
060466  005037  003100          CLR     KTFLG           ;HOLD OF KT11
060472  005737  002162          TST     TSTCNT                  ;IS THIS FIRST TEST IN RUN ?
060476  001403                  BEQ     1$                      ;CONTINUE TEST IF FIRST IN RUN
060500  012700  061673          MOV     #T40NE,R0               ;"TEST NOT EXECUTED"
060504  000402                  BR      100$                    ;JUST EXIT IF NOT
060506  012700  061740  1$:     MOV     #TST40ID,R0             ;TEST ID MESSAGE
060512  004737  017372  100$:   JSR     PC,TSTSETUP             ;DO THE COMMON SETUP
060516  004737  021366          JSR     PC,CHKMAN               ;SEE IF MANUAL INTERVENTION ALLOWED
060522  103402                  BCS     102$                    ;CARRY SET IF INTERVENTION ALLOWED
060524  000137  061156          JMP     64$                     ;EXIT IF NO MANUAL INTERVENTION
060530  022737  000001  002162  102$:  CMP     #1,TSTCNT               ;WAS THIS THE FIRST TEST IN SEQ
060536  001402                  BEQ     2$                      ;BR, IF IT WAS
060540  000137  061156          JMP     64$                     ;JMP TO END OF TEST
060544  004737  016630  2$:     JSR     PC,SOFINIT              ;DO A SOFT INIT
060550  103405                  BCS     5$                      ;BRANCH IF OK
060552  010001                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
060554                          ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
060554  104455                                                          TRAP    C$ERDF
```

```
                                                                                .WORD   801
                                                                                .WORD   SFIERR
                                                                                .WORD   SFIMSG
 9265 060564 012700 061170          5$:     MOV     #SCMENU,R0          ;MENU OF SCOPE LOOP SELECTIONS
 9266 060570 012701 000010                  MOV     #8.,R1              ;MAXIMUM ALLOWED SELECTION
 9267 060574 004737 021144                  JSR     PC,GETSEL           ;GO GET THE OPERATORS SELECTION
 9268 060600 005700                         TST     R0                  ;WAS ZERO SPECIFIED ?
 9269 060602 001760                         BEQ     2$                  ;REPEAT MENU IF YES.
 9270 060604 010004                 3$:     MOV     R0,R4               ;SAVE THE MENU SELECTION
 9271 060606                                SETPRI  #PRI07              ;RAISE THE PRIORITY
                                                                                MOV     #PRI07,R0
      060606 012700 000340                                                      TRAP    C$SPRI
      060612 104441
 9272 060614 005037 061162                  CLR     TTION               ;ASSUME INTERRUPTS ARE ENABLED
 9273 060620 032737 000100 177560           BIT     #100,@#TTICSR       ;ARE TTI INTERRUPTS ON ?
 9274 060626 001005                         BNE     4$                  ;BRANCH IF YES
 9275 060630 005237 061162                  INC     TTION               ;FLAG SET IF INTERRUPTS OFF
 9276 060634 052737 000100 177560           BIS     #100,@#TTICSR       ;ENABLE INTERRUPTS
 9277 060642 012701 000060          4$:     MOV     #TTIVEC,R1          ;START OF TTI VECTORS
 9278 060646 011137 061164                  MOV     (R1),TVECSAV        ;SAVE THE CURRENT TTI VECTOR
 9279 060652 012721 061070                  MOV     #60$,(R1)+          ;SET NEW INTERRUPT ROUTINE
 9280 060656 011137 061166                  MOV     (R1),TPRISAV        ;SAVE THE VECTOR PRIORITY
 9281 060662 012711 000340                  MOV     #PRI07,(R1)         ;USE PRIORITY SEVEN
 9282 060666                                SETPRI  #PRI00              ;LOWER INTERRUPT BR LEVEL
                                                                                MOV     #PRI00,R0
      060666 012700 000000                                                      TRAP    C$SPRI
      060672 104441
 9283 060674 006304                         ASL     R4                  ;CONVERT TO WORD OFFSET
 9284 060676 000174 060702                  JMP     @6$(R4)             ;JUMP TO PROPER LOOP
 9285 060702 060544                 6$:     .WORD   2$                  ;RETYPE THE MENU
 9286 060704 060722                         .WORD   10$                 ;TSBA READ ACCESS
 9287 060706 060730                         .WORD   15$                 ;TSSR READ ACCESS
 9288 060710 060736                         .WORD   20$                 ;TSSR WRITE ACCESS
 9289 060712 060752                         .WORD   25$                 ;TSDB HIGH BYTE WRITE ACCESS
 9290 060714 060766                         .WORD   30$                 ;TSDB LOW BYTE WRITE ACCESS
 9291 060716 061002                         .WORD   35$                 ;TSSR BYTE WRITE (SELF-TEST)
 9292 060720 061156                         .WORD   65$                 ;LEAVE THE TEST
 9293
 9294
 9295 060722                        10$:
 9296 060722 016500 177776          12$:    MOV     TSBA(R5),R0         ;READ TSBA REGISTER
 9297 060726 000775                         BR      12$                 ;LOOP UNTIL HALTED
 9298
 9299 060730                        15$:
 9300 060730 016500 000000          18$:    MOV     TSSR(R5),R0         ;READ TSSR REGISTER
 9301 060734 000775                         BR      18$                 ;LOOP UNTIL STOPPED
 9302
 9303
 9304 060736 004737 021062          20$:    JSR     PC,GETPAT           ;READ THE DATA PATTERN
 9305 060742 010001                         MOV     R0,R1               ;DATA PATTERN FOR LOOP
 9306 060744 010165 000000          22$:    MOV     R1,TSSR(R5)         ;WRITE DATA TO TSSR
 9307 060750 000775                         BR      22$                 ;LOOP
 9308
 9309
 9310 060752                        25$:
 9311 060752 004737 021062                  JSR     PC,GETPAT           ;READ THE DATA PATTERN
 9312 060756 010001                         MOV     R0,R1               ;DATA PATTERN FOR LOOP
 9313 060760 110165 177777          27$:    MOVB    R1,TSDBH(R5)        ;WRITE THE DATA TO TSDB, HIGH BYTE
 9314 060764 000775                         BR      27$                 ;LOOP UNTIL STOPPED
```

```
 9315
 9316
 9317 060766                          30$:
 9318 060766  004737  021062                JSR     PC,GETPAT          ;READ THE DATA PATTERN
 9319 060772  010001                        MOV     R0,R1              ;DATA PATTERN FOR LOOP
 9320 060774  110165  177776          32$:  MOVB    R1,TSDB(R5)        ;WRITE DATA TO TSSR, LOW BYTE
 9321 061000  000775                        BR      32$                ;LOOP UNTIL HALTED BY OPERATOR
 9322
 9323 061002  004737  021062          35$:  JSR     PC,GETPAT          ;PICK UP THE PATTERN FROM OPR
 9324 061006  010001                        MOV     R0,R1              ;STORE IN R1
 9325 061010  110165  000000          37$:  MOVB    R1,TSSR(R5)        ;WRITE BYTE TO TSSR, THIS STARTS MDIAG
 9326 061014                                DELAY   250                ;WAIT 2500US
      061014  012727  000250                                                  MOV     #250,(PC)+
      061020  000000                                                          .WORD   0
      061022  013727  002116                                                  MOV     L$DLY,(PC)+
      061026  000000                                                          .WORD   0
      061030  005367  177772                                                  DEC     -6(PC)
      061034  001375                                                          BNE     .-4
      061036  005367  177756                                                  DEC     -22(PC)
      061042  001367                                                          BNE     .-20
 9327 061044  036527  000000  000200          BIT     TSSR(R5),#SSR      ;CHECK FOR READY SET, IF A TRAP OCCURS
 9328                                                                    ;THE TSSR WAS 'NOT BACK TO THE BUS'
 9329                                                                    ;IN TIME.
 9330 061052  001356                        BNE     37$                ;BR, IF SSR WAS SET (GOOD) KEEP LOOPING
 9334 061054                                ERRDF   ERRNO,T40NSR,SFIMSG ;PRINT DEVICE FATAL ERROR MESSAGE
      061054  104455                                                          TRAP    C$ERDF
      061056  001442                                                          .WORD   802
      061060  061754                                                          .WORD   T40NSR
      061062  011656                                                          .WORD   SFIMSG
 9335 061064  000137  000200                JMP     200                ;GO TO SUPERVISOR ETC.
 9336
 9337
 9338                                ;+
 9339                                ;PROCESS CONSOLE INTERRUPTS
 9340                                ;-
 9341
 9342 061070  010046                60$:    MOV     R0,-(SP)           ;SAVE WORK REGISTER
 9343 061072  113700  177562                MOVB    @#TTIBFR,R0        ;GET THE OPERATOR INPUT
 9344 061076  042700  000200                BIC     #200,R0            ;STRIP OFF PARITY BIT
 9345 061102  122700  000015                CMPB    #15,R0             ;IS IT A CARRIAGE RETURN ?
 9346 061106  001021                        BNE     61$                ;JUST EXIT IF NOT
 9347 061110  012766  060544  000002        MOV     #2$,2(SP)          ;RETURN TO MASTER MENU
 9348 061116  005066  000004                CLR     4(SP)              ;FORCE PRIORITY ZERO
 9349 061122  013737  061164  000060        MOV     TVECSAV,@#TTIVEC   ;RESTORE SUPERVISOR VECTOR
 9350 061130  013737  061166  000062        MOV     TPRISAV,@#TTIVEC+2 ;RESTORE SUPERVISOR PRIORITY
 9351 061136  005737  061162                TST     TTION              ;ARE SUPERVISOR INTERRUPTS ENABLED ?
 9352 061142  001403                        BEQ     61$                ;BRANCH IF YES
 9353 061144  042737  000100  177560        BIC     #100,@#TTICSR      ;TURN OFF TTI INTERRUPTS
 9354 061152  012600                61$:    MOV     (SP)+,R0           ;RESTORE REGISTER
 9355 061154  000002                        RTI                        ;RETURN FROM INTERUPT
 9356
 9357 061156                        64$:
 9358 061156                        63$:
 9359 061156                        65$:    EXIT    TST                ;EXIT THE TEST
      061156  104432                                                          TRAP    C$EXIT
      061160  000664                                                          .WORD   L10075-.
 9360
```

```
                                        ;+
                                        ;LOCAL STORAGE FOR THIS TEST
                                        ;-

061162  000000                          TTION:          .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
061164  000000                          TVECSAV:        .WORD   0               ;SAVE TTI VECTOR
061166  000000                          TPRISAV:        .WORD   0               ;SAVE TTI PRIORITY


                                        ;+
                                        ;MENU FOR OPERATOR INPUT FOR SCOPE LOOPS
                                        ;-

                                                .EVEN
061170  061220  061273  061321  SCMENU: .WORD   1$,2$,3$,4$,5$,6$
061204  061472  061530  061567          .WORD   7$,8$,10$,11$,12$,0

012     123     105     1$:     .ASCIZ  <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
012     011     060     2$:     .ASCIZ  <12>'   0       Display This Menu'
011     061     011     3$:     .ASCIZ  '       1       TSBA Read Access'
011     062     011     4$:     .ASCIZ  '       2       TSSR Read Access'
011     063     011     5$:     .ASCIZ  '       3       Initialize (TSSR Write Access)'
011     064     011     6$:     .ASCIZ  '       4       TSDB High Byte Write Access'
011     065     011     7$:     .ASCIZ  '       5       TSDB Low Byte Write Access'
011     066     011     8$:     .ASCIZ  '       6       TSSR Write Byte (Self Test)'
011     067     011     10$:    .ASCIZ  '       7       Return to Diagnostic Supervisor'
000             011$:   .ASCIZ  ''
124     171     160     12$:    .ASCIZ  'Type RETURN To Stop Scope Loops'
123     164     141     T40NE:  .ASCIZ  'Stand-alone Scope Loops Not Executed'
123     143     157     TST40ID:        .ASCIZ  'Scope Loops'
123     123     122     T40NSR: .ASCIZ  'SSR Failed To Set After TSSR Write Byte And 10ms Delay'
                                .EVEN
                                ENDTST

062044                                                          L10075:
062044                                                                  TRAP    C$ETST
062044  104401
```

```
10469
10474
10480
10481
10482
10483                                   .SBTTL  HARDWARE PARAMETER CODING SECTION
10484
10485                                ;++
10486                                ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
10487                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
10488                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
10489                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
10490                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
10491                                ; WITH THE OPERATOR.
10492                                ;--
10493 066644                            BGNHRD
       066644  000015                   .WORD L10076-L$HARD/2
       066646                       L$HARD::
10494
10495 066646                            GPRMA   HPM1,0,0,160000,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
       066646  000031                   .WORD   T$CODE
       066650  066700                   .WORD   HPM1
       066652  160000                   .WORD   T$LOLIM
       066654  177776                   .WORD   T$HILIM
10496 066656                            GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
       066656  001031                   .WORD   T$CODE
       066660  066727                   .WORD   HPM2
       066662  000000                   .WORD   T$LOLIM
       066664  000776                   .WORD   T$HILIM
10497 066666                            GPRMD   HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
       066666  002032                   .WORD   T$CODE
       066670  066753                   .WORD   HPM3
       066672  000340                   .WORD   340
       066674  000000                   .WORD   T$LOLIM
       066676  000007                   .WORD   T$HILIM
10498 066700                            ENDHRD
                                        .EVEN
       066700                       L10076:
10499 066700   104  105  126  HPM1:   .ASCIZ  'DEVICE ADDRESS (TSSR) '
10500 066727   111  116  124  HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
10501 066753   111  116  124  HPM3:   .ASCIZ  'INTERRUPT PRIORITY      '
                                        .EVEN
10502
10503
```

```
10505                                           .SBTTL  SOFTWARE PARAMETER CODING SECTION
10506
10507                                   ;++
10508                                   ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
10509                                   ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
10510                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
10511                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
10512                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
10513                                   ; WITH THE OPERATOR.
10514                                   ;--
10515 067004                                    BGNSFT
       067004  000006                          .WORD L10077-L$SOFT/2
       067006                           L$SOFT::
10516 067006                                    GPRML   SPM1,0,-1,YES                   ;GET RAM DUMP FLAG
       067006  000130                          .WORD   T$CODE
       067010  067022                          .WORD   SPM1
       067012  177777                          .WORD   -1
10517 067014                                    GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
       067014  001130                          .WORD   T$CODE
       067016  067061                          .WORD   SPM4
       067020  177777                          .WORD   -1
10518                                   ;       GPRMD   SPM6,4,D,7777,0,7777,YES        ; GET LOCAL ERROR LIMIT
10519                                   ;       GPRMD   SPM7,6,D,7777,0,7777,YES        ; GET GLOBAL ERROR LIMIT
10520 067022                                    ENDSFT
                                                .EVEN
       067022                           L10077:
10521
10522
10523 067022      105     116     101   SPM1:   .ASCIZ  'ENABLE M7454 RAM DUMP ON ERROR'
10524 067061      111     116     110   SPM4:   .ASCIZ  'INHIBIT ITERATIONS      '
10525 067111      120     105     122   SPM6:   .ASCIZ  'PER TEST ERROR LIMIT    '
10526 067141      120     105     122   SPM7:   .ASCIZ  'PER UNIT ERROR LIMIT    '
10527                                           .EVEN
10528                                           .SBTTL  PATCH AREA
10529                                   ;+
10530                                   ;DISPATCH TABLE
10531                                   ;
10532                                   ; *** MOVE TO FRONT OF PROGRAM FOR RELEASE ***
10533                                   ;-
10534
10535 067172                                            DISPATCH        TESTNO
       067172  000010                                  .WORD   8
       067174                           LSDISPATCH::
       067174  023630                          .WORD   T1
       067176  033132                          .WORD   T2
       067200  034164                          .WORD   T3
       067202  035516                          .WORD   T4
       067204  041450                          .WORD   T5
       067206  052716                          .WORD   T6
       067210  056540                          .WORD   T7
       067212  060462                          .WORD   T8
10536
10537                                   ;
10538                                   ; FINALLY A GENEROUS PATCH AREA.
10539                                   ;
10540                                   ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
10541                                   ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
```

PATCH AREA

```
10542                                   ;
10543
10544 067214                            PATCH::
10545                                           .EVEN
10546                                   ;       .IF     NZ,..&377
10547                                   ;       .=.!377+1
10548                                   ;       .ENDC
10549 067214                                    LASTAD          ;SET LAST USED ADDRESS.
                                                .EVEN
      067214  067232                            .WORD T$FREE
      067216  000005                            .WORD T$SIZE
      067220                            L$LAST::
10550                                           .SBTTL  HARD CODED P-TABLE
10551                                   ;++
10552                                   ;       DIAGNOSTIC IS PRE-PARAMETERIZED PER THIS TABLE
10553                                   ;--
10554 067220                                    BGNSETUP        1
10555 067220                                    BGNPTAB
      067220  000000                            .WORD   0
      067222  000003                            .WORD   L10102-./2-1
      067224                            L10100:
10556 067224  172522                            .WORD           172522
10557 067226  000224                            .WORD           224
10558 067230  000240                            .WORD           PRI05
10559 067232                                    ENDPTAB
      067232                            L10102:
10560 067232                                    ENDSETUP
10561
10562         000001                            .END
```

SYMBOL TABLE

```
ADDSSR  011762 G
ADR    = 000020 G
AMBTSS  006325
ASSEMB= 000010
A1716  = 000003
BADDAT  003110 G
BADSSR  016534 G
BAR    = 174402
BENBSW  002174 G
BIE    = 040000
BIT0   = 000001 G
BIT00  = 000001 G
BIT01  = 000002 G
BIT02  = 000004 G
BIT03  = 000010 G
BIT04  = 000020 G
BIT05  = 000040 G
BIT06  = 000100 G
BIT07  = 000200 G
BIT08  = 000400 G
BIT09  = 001000 G
BIT1   = 000002 G
BIT10  = 002000 G
BIT11  = 004000 G
BIT12  = 010000 G
BIT13  = 020000 G
BIT14  = 040000 G
BIT15  = 100000 G
BIT2   = 000004 G
BIT3   = 000010 G
BIT4   = 000020 G
BIT5   = 000040 G
BIT6   = 000100 G
BIT7   = 000200 G
BIT8   = 000400 G
BIT9   = 001000 G
BOE    = 000400 G
BRINIT  004355
BSEL0  = 000000
BSEL1  = 000001
CHKAMB  016700
CHKMAN  021366 G
CHKTSS  017220
CKDROP  020136
CKEMAX  017764
CKMSG   011202 G
CKMSG2  011322 G
CKRAM   010524 G
CKRAM2  011100 G
CMPMEM  020542
CONFIG  020204
COUNT   002254 G
CSR    = 174400
CSRADD  002154 G
CTAB    003116 G
CTABE   003130 G
CTABM   003116 G
C$AU   = 000052
C$AUTO= 000061
C$BRK  = 000022
C$BSEG= 000004
C$BSUB= 000002
C$CEFG= 000045
C$CLCK= 000062
C$CLEA= 000012
C$CLOS= 000035
C$CLP1= 000006
C$CVEC= 000036
C$DCLN= 000044
C$DODU= 000051
C$DRPT= 000024
C$DU   = 000053
C$EDIT= 000003
C$ERDF= 000055
C$ERHR= 000056
C$ERRO= 000060
C$ERSF= 000054
C$ERSO= 000057
C$ESCA= 000010
C$ESEG= 000005
C$ESUB= 000003
C$ETST= 000001
C$EXIT= 000032
C$GETB= 000026
C$GETW= 000027
C$GMAN= 000043
C$GPHR= 000042
C$GPLO= 000030
C$GPRI= 000040
C$INIT= 000011
C$INLP= 000020
C$MANI= 000050
C$MEM  = 000031
C$MSG  = 000023
C$OPEN= 000034
C$PNTB= 000014
C$PNTF= 000017
C$PNTS= 000016
C$PNTX= 000015
C$QIO  = 000377
C$RDBU= 000007
C$REFG= 000047
C$RESE= 000033
C$REVI= 000003
C$RFLA= 000021
C$RPT  = 000025
C$SEFG= 000046
C$SPRI= 000041
C$SVEC= 000037
C$TPRI= 000013
DAR    = 174404
DATA    002256 G
DATAFL  015047
DATASC  021120
DEBUGM  011454
DEVCNT  002166 G
DEVDRO  023560
DEVNRD  023477
DEVNXR  023415
DEVONL  023345
DEVSUM  023310
DFPTBL  002124 G
DIAGMC= 000000
DLCYL  = 000177
DLDNER= 100200
DLERR  = 177730
DLGETS= 000004
DLRDHD= 000010
DLRDNH= 000016
DLSR   = 000013
DLUN   = 000006
DSBINT  017040
DUAD12  004541
DUFLG   003060 G
DUMMY   003030
EF.CON= 000036 G
EF.NEW= 000035 G
EF.PWR= 000034 G
EF.RES= 000037 G
EF.STA= 000040 G
EMAXDU  017661
EN     = 000000
ENAINT  017006
ENVIRN  021516
EPRTSW  002146 G
EPRT1   005755
EPRT2   005672
EPRT3   006014
ERCM    011555
ERRHI   002202 G
ERRK    017640
ERRLO   002204 G
ERRNO  = 001442
ERRVEC= 000004 G
ERTABE  003330
ERTABL  003130
ESUM    017642
EVL    = 000004 G
EXBCNT= 000010
EXPBRE  016336 G
EXPD    002176 G
EXPGOT  004431
EXPGT2  004465
EXPMSG  002266 G
EXPREC  016330 G
EXTA    005232
EXTEND  005230
ESEND  = 002100
ESLOAD= 000035
FATCHK  020064
FATERR= 000060
FATFLG  002170 G
FERCM   011544
FIFEXP  012012 G
FIF1MS  012064
FIF2MS  012133
FILLME  020356
FNOINT  004113
FORCER  002144 G
FREE    003072 G
FREEHI  003076
FRESIZ  003074 G
FUSI    004015
F$AU   = 000015
F$AUTO= 000020
F$BGN  = 000040
F$CLEA= 000007
F$DU   = 000016
F$END  = 000041
F$HARD= 000004
F$HW   = 000013
F$INIT= 000006
F$JMP  = 000050
F$MOD  = 000000
F$MSG  = 000011
F$PROT= 000021
F$PWR  = 000017
F$RPT  = 000012
F$SEG  = 000003
F$SOFT= 000005
F$SRV  = 000010
F$SUB  = 000002
F$SW   = 000014
F$TEST= 000001
GDDAT   003112 G
GERRMA  002142 G
GETPAT  021062 G
GETSEL  021144 G
G$CNTO= 000200
G$DELM= 000372
G$DISP= 000003
G$EXCP= 000400
G$HILI= 000002
G$LOLI= 000001
G$NO   = 000000
G$OFFS= 000400
G$OFSI= 000376
G$PRMA= 000001
G$PRMD= 000002
G$PRML= 000000
G$RADA= 000140
G$RADB= 000000
G$RADD= 000040
G$RADL= 000120
G$RADO= 000020
G$XFER= 000004
G$YES  = 000010
HIADDR= 001400
HIMEM  = 007776
HOE    = 100000 G
HPM1    066700
HPM2    066727
HPM3    066753
IBE    = 010000 G
IDU    = 000040 G
IER    = 020000 G
IFAULT  004154
INCERK  017726
INTCPC  017004
INTFLA  017001
INTMAS  017000
INTR    017052 G
INTREC  002172 G
INTVEC  017002
INTX    004176
IOKCKI= 000200
IOKSTP= 000001
IPRI    002160 G
ISR    = 000100 G
IVEC    002156 G
IXE    = 004000 G
I$AU   = 000041
I$AUTO= 000041
I$CLN  = 000041
I$DU   = 000041
I$HRD  = 000041
I$INIT= 000041
I$MOD  = 000040
I$MSG  = 000041
I$PROT= 000040
I$PTAB= 000041
I$PWR  = 000041
I$RPT  = 000041
I$SEG  = 000041
I$SETU= 000041
I$SFT  = 000041
I$SRV  = 000041
I$SUB  = 000041
I$TST  = 000041
J$JMP  = 000167
KIPAR0= 172340
KIPAR1= 172342
KIPAR2= 172344
KIPAR3= 172346
KIPAR4= 172350
KIPAR5= 172352
KIPAR6= 172354
KIPAR7= 172356
KIPDR0= 172300
KIPDR1= 172302
KIPDR2= 172304
KIPDR3= 172306
KIPDR4= 172310
KIPDR5= 172312
KIPDR6= 172314
```

SYMBOL TABLE

```
KIPDR7= 172316          L$SOFT  067006 G        L10061  043530          NXRERR  005176 G        O.GO2   063750
KTENAB  003102 G        L$SPC   002056 G        L10062  044166          NXRX    003675          O.HIGH  066546
KTFLG   003100 G        L$SPCP  002020 G        L10063  044614          NXTU    022210          O.LG  = 000010
KTINIT  021604          L$SPTP  002024 G        L10064  045242          OFL   = 000100          O.LGCH  066141
KTOFF   020230          L$STA   002030 G        L10065  045624          ONEFIL= 000001          O.LGDR  062536
KTON    020212          L$SW    002134 G        L10066  046300          O$APTS= 000000          O.LOW   066544
LERRMA  002140 G        L$TEST  002114 G        L10067  046542          O$AU  = 000001          O.MOVE  064250
LISTAL= 000001          L$TIML  002014 G        L10070  047034          O$BGNR= 000001          O.MSK   066542
LOE   = 040000 G        L$UNIT  002012 G        L10071  047324          O$BGNS= 000001          O.ODT   062046 G
LOOPCN  002164 G        L10000  002132          L10072  050000          O$DU  = 000001          O.OFST  063366
LOOPCO  012750          L10001  002144          L10073  056536          O$ERRT= 000000          O.OLD   062764
LOOPFL  003114 G        L10002  005226          L10074  060460          O$GNSW= 000001          O.OP1   062770
LOT   = 000010 G        L10003  011666          L10075  062044          O$POIN= 000001          O.OP2   063034
L$ACP   002110 G        L10004  011716          L10076  066700          O$SETU= 000001          O.OP2A  063042
L$APT   002036 G        L10005  011734          L10077  067022          O.ADR1  066556          O.ORAB  062274
L$AU    022544 G        L10006  011742          L10100  067224          O.ALL   065142          O.ORPC  062252
L$AUT   002070 G        L10007  011760          L10102  067232          O.AS    062636          O.ORRB  062304
L$AUTO  022750 G        L10010  011776          MEMADD  013576 G        O.ASC   066125          O.P     066121
L$CCP   002106 G        L10011  012010          MENASC  021335          O.ASCI  064152          O.PCS   062264
L$CLEA  023024 G        L10012  012062          MENERR  021262          O.BACK  063122          O.PRNT  064410
L$CO    002032 G        L10013  012232          MENRES  021364          O.BALL  065026          O.PROC  063766
L$DEPO  002011 G        L10014  012746          MESBFA  002716 G        O.BD    066126          O.PROM  066134
L$DESC  003342 G        L10015  013574          MESBFN  014617          O.BKP = 000016          O.RALL  063312
L$DESP  002076 G        L10016  013616          MESHEA  015002          O.BKPT  063150          O.RCSR= 177560
L$DEVP  002060 G        L10017  016334          MIMENU  055552          O.BRK   064456          O.RDB = 177562
L$DISP  067174 G        L10020  016342          MMVEC = 000250          O.BW    066106          O.REG   066040
L$DLY   002116 G        L10021  016350          MPR   = 174406          O.BYT   062674          O.REGT  062164
L$DTP   002040 G        L10022  016362          MSA.FR= 000006          O.BYT1  062666          O.REM   065312
L$DTYP  002034 G        L10023  016404          MSA.NO= 000000          O.CAD   066110          O.RSB   065246
L$DU    022642 G        L10024  016432          MSA.NR= 000004          O.CADV  065454          O.RSR   065216
L$DUT   002072 G        L10025  016572          MSA.VO= 000002          O.CLGT= 000035          O.RSTT  065406
L$DVTY  003334 G        L10026  017102          MSGEXP  012000 G        O.CLSE  065752          O.S     066117
L$EF    002052 G        L10030  022474          MSGLOO  012706 G        O.COMP  064312          O.SCAN  062430
L$ENVI  002044 G        L10031  022640          MSGSTA  012172 G        O.CR    066131          O.SEMI  062630
L$ETP   002102 G        L10032  022746          MSGSUB  013564 G        O.CRET  062756          O.SEQ   066124
L$EXP1  002046 G        L10033  023022          MS.ATT= 000006          O.CRLF  066004          O.SNGL  062354
L$EXP4  002064 G        L10034  023044          MS.EXT= 000200          O.CRLS  066020          O.SPAC  065740
L$EXP5  002066 G        L10035  023306          MS.RSD= 000001          O.CSR1  066122          O.STM = 000340
L$HARD  066646 G        L10036  033130          MS.RSF= 000020          O.CSR2  066123          O.SVR   065156
L$HIME  002120 G        L10037  024260          MS.RST= 000010          O.CT    066600          O.SVTT  065360
L$HPCP  002016 G        L10040  025304          NBA   = 002000          O.C1    064040          O.SWCH  066550
L$HPTP  002022 G        L10041  026460          NEWPAS  022176          O.DCD   062404          O.T     066120
L$HW    002124 G        L10042  026762          NODEV   003062 G        O.DCDA  062762          O.TBIT  063716
L$ICP   002104 G        L10043  030034          NOINIT  004233          O.DCDB  063310          O.TBT = 000020
L$INIT  021746 G        L10044  030734          NOINTR  004117          O.DCD1  062424          O.TCLS  062326
L$LADP  002026 G        L10045  034162          NOITS   002136 G        O.DCD2  062420          O.TCSR= 177564
L$LAST  067220 G        L10046  035514          NOMAN   021422          O.DOT   066112          O.TDB = 177566
L$LOAD  002100 G        L10047  034424          NP.IR = 000200          O.DUMP  064072          O.TL    066176
L$LUN   002074 G        L10050  041446          NP.LOO= 000040          O.EFF   063476          O.TRTC  066206
L$MREV  002050 G        L10051  036114          NP.OUT= 000100          O.ERR   062374          O.TVEC= 000014
L$NAME  002000 G        L10052  036572          NP.WRP= 000020          O.ERR1  063472          O.TYPE  065724
L$PRIO  002042 G        L10053  037000          NSI     004050          O.FCHR  066552          O.UIN   066622
L$PROT  021736 G        L10054  037266          NSINIT  004305          O.FCNT  066554          O.UPC   066536
L$PRT   002112 G        L10055  037610          NUL     004425          O.FTYP  065570          O.UPS   066540
L$REPP  002062 G        L10056  052714          NULCR   004426          O.GET   065636          O.URO   066520
L$REV   002010 G        L10057  042266          NXM   = 004000          O.GO    063666          O.USP   066534
L$RPT   023046 G        L10060  043034          NXR     003636          O.GO1   063744          O.WB1   062702
```

SYMBOL TABLE

```
O.WDFG  066116          PST32W  003104 G        REWIND  010424 G        S1.IFM= 001000          TST21I  034006
O.WRD   062652          PUNIT   022476          RMCHBE= 000167          S1.IHE= 000400          TST22I  035322
O.WRD1  062716          PW.D11= 000021          RMCHEN= 000200          S1.IID= 004000          TST23I  041251
O.WSCH  063502          PW.D13= 000022          RMMSGB= 000104          S1.I1R= 020000          TST24I  052462
O.XXX   066114          PW.D22= 000020          RMMSGE= 000117          S1.I2R= 040000          TST39I  060432
PASRPT  022242          PW.NOP= 000000          RMPKTB= 000020          S1.PAR= 100000          TST40I  061740
PATCH   067214 G        PW.NO1= 000023          RMPKTE= 000027          S2.ATI= 000010          TTIBFR= 177562 G
PATDAT  021116          PW.RDE= 000024          RMR   = 010000          S2.BTI= 000004          TTICSR= 177560 G
PC.ERA= 002400          PW.RDR= 000001          RWPACK  010520          S2.DIM= 000200          TTION   061162
PC.IER= 002000          PW.RDS= 000005          SC    = 100000          S2.ILW= 000100          TTION2  053750
PC.NOO= 001000          PW.RFI= 000003          SCE   = 020000          S2.INR= 000020          TTIVEC= 000060 G
PC.REL= 000000          PW.WCT= 000006          SCME    004711          S2.OUT= 000040          TTOBFR= 177566
PC.REW= 000400          PW.WFI= 000004          SCMENU  061170          S2.UND= 000003          TTOCSR= 177564
PKBCNT= 000006          PW.WFM= 000007          SDELAY  010320          TBLEND= 003030 G        TUV2A   002000 G
PKHI  = 000004          PW.WMI= 000010          SEEK  = 000006          TCOASC  006166          TVECSA  061164
PKLOW = 000002          PW.WNP= 000011          SELASC  021330          TCOCOD  006366          TVSAV2  053752
PKTADD  007264          PW.WTR= 000002          SELDAT= 000004          TEMP1   003064 G        T$ARGC= 000001
PKTFRM  007226          P.ACK = 100000          SEL2  = 000002          TEMP2   003066 G        T$CODE= 001130
PKTGET  011720 G        P.CMD = 000037          SETMAP  020252          TERCLS= 000016          T$ERRN= 001442
PKTMES  011744 G        P.CONT= 000012          SETU    022274          TESTNO= 000010          T$EXCP= 000000
PKTNEW  007321          P.CVC = 040000          SFFMSG  011736 G        TEXASC  006125          T$FLAG= 000040
PKTRAM  004643 G        P.FMT = 000140          SFHERR  003603          TFCASC  006227          T$FREE= 067232
PKTSSR  011670 G        P.FORM= 000011          SFIERR  003550          TIMEXP  016406 G        T$GMAN= 000000
PNT   = 001000 G        P.GETS= 000017          SFIMSG  011656 G        TIMSGO  016434          T$HILI= 000007
PRAMPK  013620          P.IE  = 000200          SFPTBL  002134 G        TINERR  011643          T$LAST= 000001
PRBEXP  016324          P.INIT= 000013          SIFLAG  003106 G        TKB   = 177562          T$LOLI= 000000
PRBMSG  016172          P.MODE= 007400          SIMSG   011610          TKS   = 177560          T$LSYM= 010000
PRBREC  016326          P.OPP = 020000          SKIP    037610          TMPBFR  002576 G        T$LTNO= 000010
PRBTOT  016257          P.POSI= 000010          SKIPT   003332          TMPEND  030762          T$NEST= 000000
PRBYTE  015756 G        P.READ= 000001          SOFINI  016630 G        TNAM    017566          T$NS0 = 000000
PRI   = 002000 G        P.SWB = 010000          SPACE   010126 G        TPB   = 177566          T$NS1 = 000005
PRIADD  007700          P.WRIT= 000005          SPM1    067022          TPRISA  061166          T$NS2 = 000002
PRIAO   007750          P.WRTC= 000004          SPM4    067061          TPS   = 177564          T$PCNT= 000000
PRIBXO  007332 G        P.WRTS= 000006          SPM6    067111          TPSAV2  053754        T$PTAB= 010101
PRIEQU  007600          QVP     002152 G        SPM7    067141          TRANST  002134 G        T$PTHV= 000001
PRIPKT  007060 G        RAMASC  013766          SR0   = 177572          TSBA  = 177776 G        T$PTNU= 000001
PRIRAM  007606          RAMDAT  002206 G        SR1   = 177574          TSBAH = 177777 G        T$SAVL= 177777
PRITAD  010014          RAMER   010626 G        SR2   = 177576          TSBAL = 177776 G        T$SEGL= 177777
PRITSS  005264          RAMERR  016344 G        SR3   = 172516          TSDB  = 177776 G        T$SIZE= 000005
PRITO   010064          RAMEXP  016364 G        SSR   = 000200          TSDBH = 177777 G        T$SUBN= 000000
PRIXOR  007462 G        RAMFHR  014526          STATCO  012234          TSDBL = 177776 G        T$TAGL= 177777
PRI00 = 000000 G        RAMFOR  007636          SVCGBL= 000000          TSFCOD  006726          T$TAGN= 010103
PRI01 = 000040 G        RAMHLD  011010          SVCINS= 000000          TSREJ = 000006          T$TEMP= 000011
PRI02 = 000100 G        RAMIOP  011014          SVCSUB= 000001          TSSDEF  006276          T$TEST= 000010
PRI03 = 000140 G        RAMPD   011065          SVCTAG= 000000          TSSR  = 000000 G        T$TSTM= 177777
PRI04 = 000200 G        RAMR5H  011012          SVCTST= 000001          TSSRBI  003400 G        T$TSTS= 000001
PRI05 = 000240 G        RAMSIZ  002246 G        S$LSYM= 010000          TSSRFO  006105          T$$AU = 010031
PRI06 = 000300 G        RAMTAD  016352 G        S0.IDB= 000010          TSSRH = 000001 G        T$$AUT= 010033
PRI07 = 000340 G        RBPCRA  015114          S0.IFB= 000002          TSSX    003716          T$$CLE= 010034
PRMESS  014052          RCVHIA  002250 G        S0.IFP= 000001          TSTBLK  002720 G        T$$DAT= 010102
PRMNO   002264 G        RCVLOA  002252 G        S0.ILD= 000020          TSTCNT  002162 G        T$$DU = 010032
PRMSGE  015406 G        RDERR   005104          S0.ION= 000040          TSTEND  017602          T$$HAR= 010076
PRMSGO  015566          READ  = 000014          S0.IRD= 000100          TSTFLA  002260 G        T$$HW = 010000
PRMSG1  015633          READY = 000001          S0.IRW= 000004          TSTLOO  017340 G        T$$INI= 010030
PRMSG2  015671          RECMSG  002432 G        S0.ISP= 000200          TSTPTR  002262 G        T$$MSG= 010025
PROASC  014677          RECV    002200 G        S1.ICE= 002000          TSTSET  017372 G        T$$PC = 000001
PR1ASC  014744          REGSAV  021022          S1.IEO= 010000          TST17I  031216          T$$PRO= 010027
```

SYMBOL TABLE

```
T$$PTA= 010101          T21DLY  033566          T23TM   040226          T3.1    034224          T39TAD  057230
T$$RPT= 010035          T21LOO  033172          T23TMA  040315          T38ASC  056471          T39WPN  060234
T$$SOF= 010077          T21OFL  033765          T23VCK  041035          T38ASN  056510          T39WR   057762
T$$SRV= 010026          T21PAC  033440          T23WB   040002          T38ASO  056366          T39WRT  060161
T$$SUB= 010072          T21PK2  033550          T23WD   040020          T38AS1  056433          T4      035516 G
T$$SW = 010001          T21RES  034030          T23WDC  040733          T38BFR  054006          T4.1    035566
T$$TES= 010075          T21RT2  034120          T23WDD  040644          T38BSO  054000          T4.2    036132
T1      023630 G        T21SSR  033570          T23WDR  040022          T38BS1  054001          T4.3    036610
T1.1    023666          T21S2   033562          T23WSS  041162          T38BS2  054002          T4.4    037016
T1.2    024274          T21S3   033564          T24AM3  051450          T38CNT  056534          T4.5    037304
T1.3    025320          T22AM3  034725          T24BA   052002          T38DAT  056024          T40NE   061673
T1.4    026474          T22BFR  034512          T24BFR  050060          T38DLY  053756          T40NSR  061754
T1.5    026776          T22BF2  034610          T24BF2  050200          T38DTA  054500          T5      041450 G
T1.6    030050          T22BSO  034610          T24BOT  051043          T38EAI  054506          T5.1    041514
T17BEN  033002          T22BS1  034611          T24BSO  050200          T38EB   054462          T5.10   046560
T17BFR  032662          T22DAT  034500          T24BS1  050201          T38ID   055526          T5.11   047052
T17BFS  032702          T22FOR  034624          T24CON  050212          T38MBP  056064          T5.12   047342
T17CLE  032564          T22LOO  034224          T24DAT  050050          T38MSG  055430          T5.2    042304
T17CLR  032376          T22OFL  035025          T24DLY  050216          T38MS2  055364          T5.3    043052
T17DAT  032650          T22PAC  034470          T24DTA  051110          T38MS4  055144          T5.4    043546
T17DLY  033126          T22PK2  034600          T24EOT  051176          T38MS5  055223          T5.5    044204
T17DT2  033020          T22POS  034622          T24ILA  050572          T38NE   054550          T5.6    044632
T17EXE  031114          T22RD   034616          T24LON  052142          T38OFL  055110          T5.7    045260
T17EXP  030772          T22RES  035356          T24LOO  041514          T38ONL  055046          T5.8    045642
T17EXS  031012          T22RT2  035450          T24LOP  052224          T38PAC  053770          T5.9    046316
T17LOO  024046          T22RWJ  035174          T24LOQ  050656          T38PK2  054470          T6      052716 G
T17MSK  030766          T22SSR  034630          T24LOR  050272          T38PK3  054520          T7      056540 G
T17PAC  032640          T22S2   034612          T24NEF  050220          T38PK4  054540          T8      060462 G
T17PK2  033010          T22S3   034614          T24NXM  050431          T38RES  056026          UAM   = 000200 G
T17RFI  032544          T22TM   035100          T24OFL  051515          T38SIZ  054546          UNITN   002150 G
T17RSF  032442          T22VCK  035247          T24PAC  050040          T38SST  055300          UNREC = 000006
T17RWN  031235          T22WRT  034620          T24PBP  052306          T38TAD  054000          USI     004021
T17SET  032604          T23AM3  040523          T24PK2  050150          T38WLE  054752          WAITF   017104 G
T17SNP  032464          T23BA   041110          T24PK3  050170          T38WOK  055470          WC.IFA= 000200
T17SRD  032422          T23BFR  037670          T24RB   050172          T38WR   054542          WC.IFE= 000002
T17SSR  031263          T23BF2  040012          T24RES  052530          T38WRL  054711          WC.IGO= 000001
T17WFD  031114          T23BSO  040012          T24RN   050206          T38WRT  054625          WC.IRE= 000010
T17WFI  032510          T23BS1  040013          T24RNC  051375          T39BFR  057236          WC.IRW= 000004
T171CM  031603          T23CON  040024          T24RT2  052622          T39BSO  057230          WC.IOT= 000100
T172CM  031665          T23DAT  037660          T24RT3  052664          T39BS1  057231          WC.I1T= 000040
T172SS  031320          T23EOT  040150          T24RWN  051326          T39BS2  057232          WC.I5R= 000020
T173CM  031761          T23ET   040063          T24SSR  050737          T39DAT  060372          WF.IED= 000010
T173SS  031364          T23LOO  035566          T24SZ   050176          T39DLY  057206          WF.IER= 000004
T174CM  032045          T23OFL  040571          T24S2   050202          T39DTA  057730          WF.IHI= 000200
T174SS  031431          T23PAC  037650          T24S3   050204          T39EAI  057736          WF.IRE= 000040
T175CM  032130          T23PK2  037760          T24TM   051253          T39MCL  060313          WF.IWF= 000020
T175SS  031474          T23PK3  040000          T24TRL  052374          T39NE   057773          WF.IWR= 000100
T176CM  032204          T23RES  041266          T24VCK  051727          T39NFL  057770          WF.I3R= 000002
T176SS  031540          T23RNC  040450          T24WB   050172          T39OF2  060052          WF.I4R= 000001
T177CM  032312          T23RSZ  040010          T24WDC  051656          T39ON2  060116          WRTCHR  010322 G
T2      033132 G        T23RT2  041360          T24WDD  051570          T39PAC  057220          WRTERR  005011
T21AM3  033665          T23RT3  041422          T24WDE  050771          T39PK2  057720          WRTMSG  004754
T21BFR  033460          T23RWN  040401          T24WDF  050515          T39PK3  057750          XFERAS  016574
T21BF2  033560          T23SSR  040030          T24WDG  050342          T39PK4  057760          XNXM    017260
T21BSO  033560          T23SZ   040006          T24WDR  050210          T39RES  060374          XORBFO  007414
T21BS1  033561          T23S2   040014          T24WSS  052053          T39RL   060370          XORFOR  007532
T21DAT  033450          T23S3   040016          T3      034164 G        T39SIZ  057766          XSIO  = 000006 G
```

```
CZTUXA0 TU80 FRONT END PRT B    MACRO M1200  29-MAR-83 13:32  PAGE 141-6
SYMBOL TABLE

XST1   = 000010 G       XSONEF= 002000          XS3CON  015340          X1.UNC= 000002          X3.MDE= 177400
XST2   = 000012 G       XSOONL= 000100          XXCOMM  003070 G        X2.BUF= 000100          X3.OPI= 000100
XST3   = 000014 G       XSOPED= 000010          X$ALWA= 000000          X2.EXT= 000200          X3.REV= 000040
XST4   = 000016 G       XSORLL= 010000          X$FALS= 000040          X2.OPM= 100000          X3.RIB= 000001
XSOBOT= 000002          XSORLS= 040000          X$OFFS= 000400          X2.RCE= 040000          X3.SPA= 000200
XSOCON  015161          XSOTMK= 100000          X$TRUE= 000020          X2.REV= 000077          X3.TRF= 000020
XSOEOT= 000001          XSOVCK= 000020          X1.COR= 020000          X2.SPA= 035400          X4.HSP= 100000
XSOIE  = 000040         XSOWLE= 004000          X1.DLT= 100000          X2.UNI= 000007          X4.MBZ= 017400
XSOILA= 000400          XSOWLK= 000004          X1.MBZ= 017375          X2.WCF= 002000          X4.RCE= 040000
XSOILC= 001000          XS1CON  015226          X1.RBP= 000400          X3.DCK= 000010          X4.TSM= 020000
XSOLET= 020000          XS2CON  015273          X1.SPA= 040000          X3.MBZ= 000006          X4.WRC= 000377
XSOMOT= 000200

. ABS.  067232     000
        000000     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  36936 WORDS  ( 145 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:09:15
CZTUXA.BIC,CZTUXA/-SP=SVC.MLB/ML,CZTUXA.MAC
```