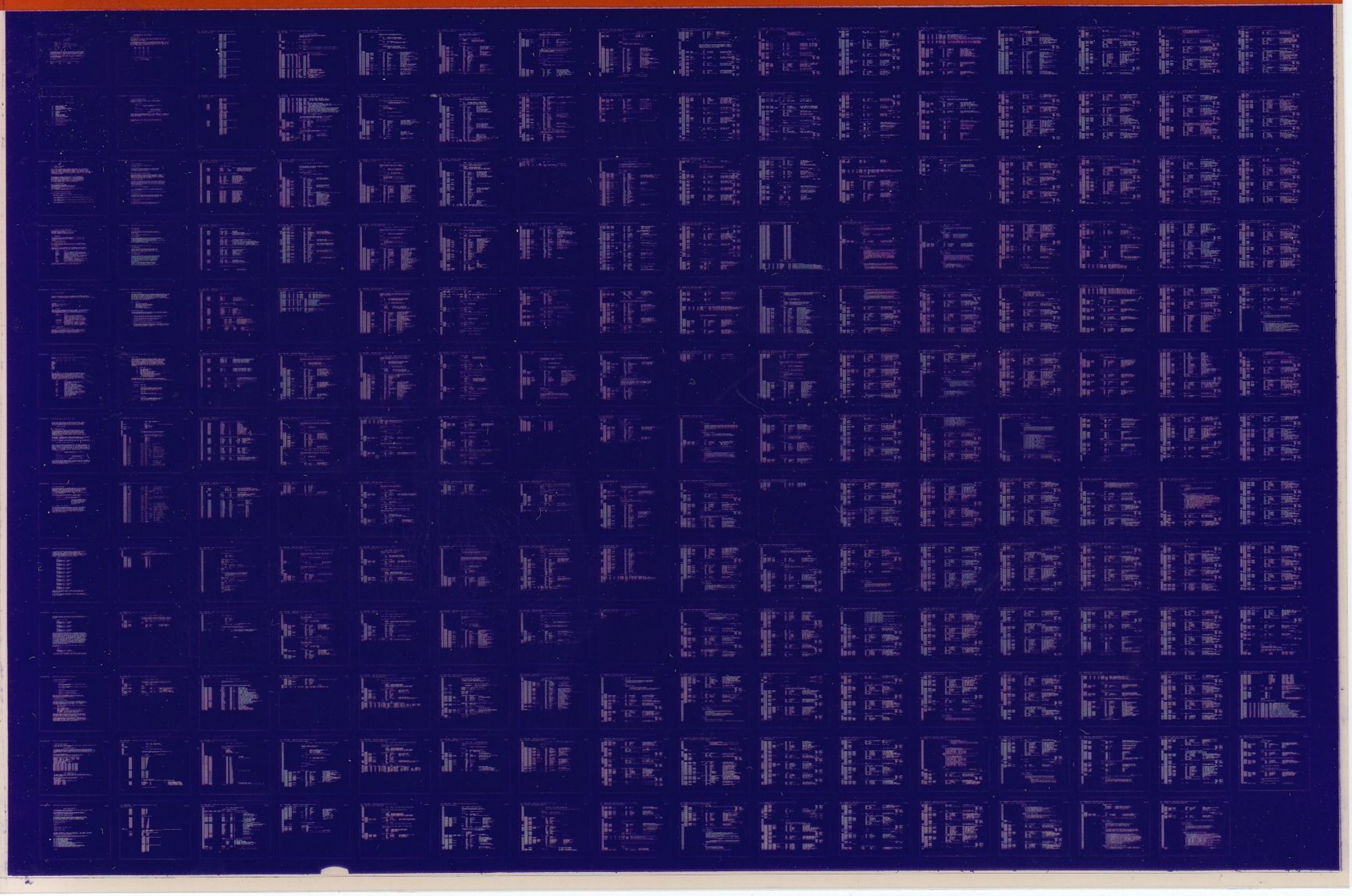
TSV05

TSV05 CONTROL PART 2
CVTSBD0

AH-T096D-MC
1 OF 2 JAN 1986
 MADE IN USA

digital

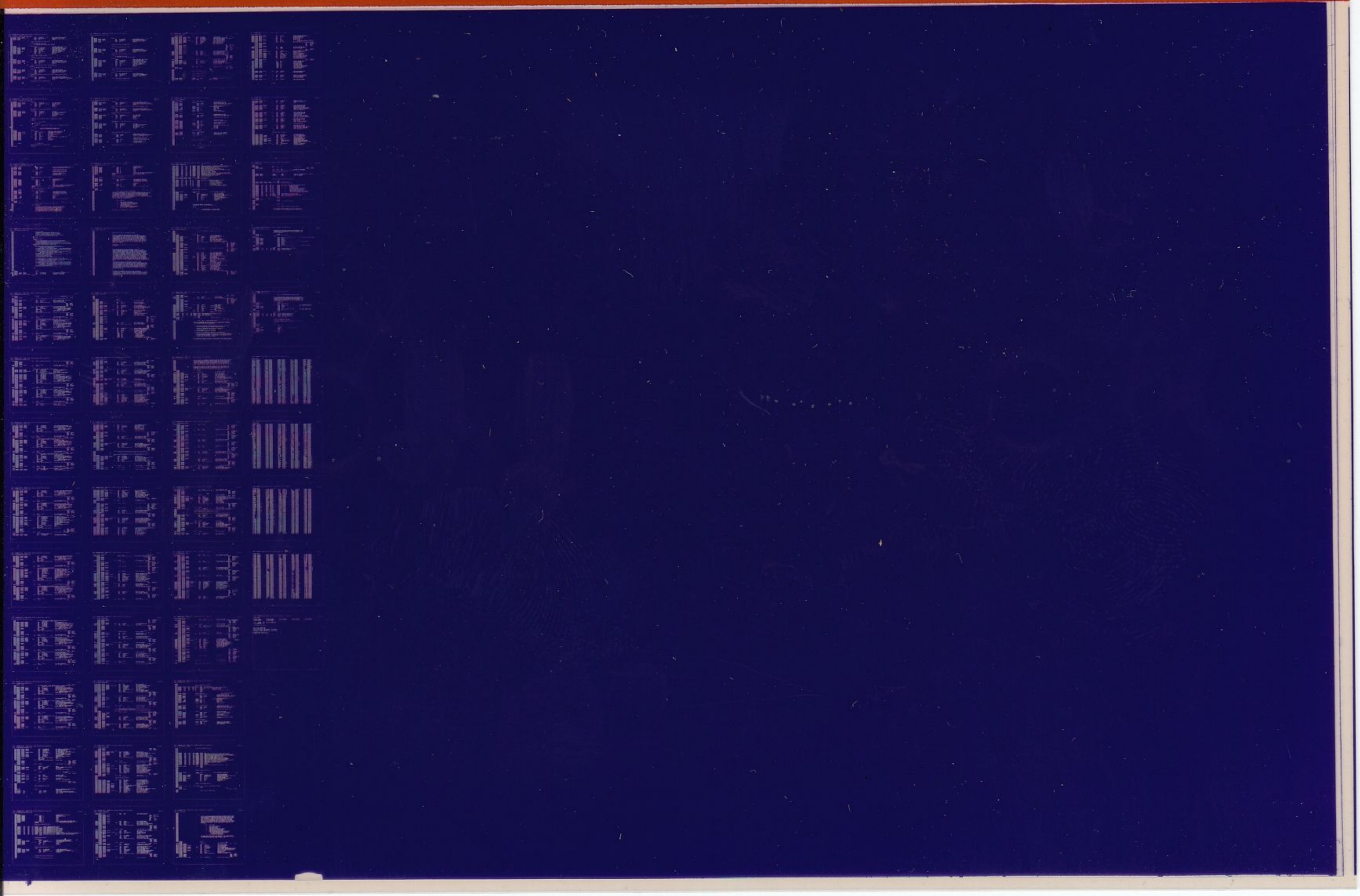


TSV05

TSV05 CONTROL PART 2
CVTSBD0

AH-T096D-MC
2 OF 2 JAN 1986
 MADE IN USA

digital



.REM_

IDENTIFICATION

PRODUCT ID: AC-T095D-MC

PRODUCT TITLE: CVTSBDO TSV05 CTRL PART 2

DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PGG

DATE: AUGUST 23, 1985

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS IS A PDP-11/23 RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A PDP-11//23 SYSTEM (Q-BUS). THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. THIS DIAGNOSTIC CONSIST OF TWELVE TEST. TEST 1-9 ARE EXECUTED IN SEQUENCE. TEST 10-12 ARE STAND ALONE TEST WHICH ALLOW THE OPERATOR TO PERFORM SPECIFIC FUNCTIONAL TEST ON SCOPE LOOPS ON CERTAIN FUNCTIONS.

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP+, ACT, APT, SLIDE AND PAPER TAPE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP+ USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

PDP-11/23 PROCESSOR AND MEMORY
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE AND 4K RESERVED FOR I/O PAGE)
TSV05 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CONSOLE TERMINAL
PDP-11 DIAGNOSTIC SUPERVISOR (HSAAA.SYS VERSION 34 OR LATER)
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. CHQUS XXDP+ USERS GUIDE; DOCUMENT NUMBER AC-F348E-MC
   DATE: 14 JULY 1980.

2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE; DOCUMENT NUMBER EK-TSV05-UG-001
   DATE: AUGUST 1982

3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSV05-TM-001
   DATE: AUGUST 1982

4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSV05-IN-001
   DATE: AUGUST 1982

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

FUNCTIONAL PDP-11/23 CENTRAL PROCESSOR AND MEMORY
FUNCTIONAL CONSOLE TERMINAL
FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR
FUNCTIONAL DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK PROPERLY OR FALSE ERRORS CAN BE REPORTED.
THE TAPE BEING USED ON THE TS05 TRANSPORT IS A KNOWN GOOD REEL OF TAPE.
CVTSAA HAS RUN SUCESSFULLY.

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES. FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER ↑C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

2.1.1 OPERATOR COMMANDS

THE TSV05 DIAGNOSTIC IS A PDP-11/23 DIAGNOSTIC SUPERVISOR COMPATIBLE PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE CHQUS XXDP+ USERS GUIDE,DOCUMENT NUMBER AC-F348E-MC. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC MEDIA

```
.R VTSB??
DIAG. RUN-TIME SERVICES REV D. APR 79
CVTSB-A-0
****TSV05 LOGIC DIAGNOSTIC****
UNIT IS TSV05
>DR
```

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

```
START/TESTS:1-5/PASS:1000/EOP:100
```

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TEST#	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A "BELL" ON ERROR, YOU MAY USE THE FOLLOWING STRING:

```
/FLAGS:LOE:IER:BOE
```

2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?" YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 14 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL).

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 RUN USING THE DEFAULT VALUES FOR ALL QUESTIONS. THE DEFAULT ADDRESS AND VECTOR ARE:

```
TSBA/TSDB = 172520, 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 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 M7196 CONTROLLERS
                PRESENT TO BE TESTED>

UNIT 0

DEVICE ADDRESS  (O) 172520 ? <ENTER THE ADDRESS OF THE
                              TSBA/TSDB 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 AS FOLLOWS:
UP TO 4 TSV05 CONTROLLERS PER 11/23 AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

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.>
```

2.6 EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH UNIT TO BE TESTED. IF YOU HAVE A MULTIPLEXED DEVICE SUCH AS A MASS STORAGE CONTROLLER WITH SEVERAL DRIVES OR A COMMUNICATION DEVICE WITH SEVERAL LINES, THIS BECOMES TEDIOUS SINCE MOST OF THE ANSWERS ARE REPETITIOUS.

TO ILLUSTRATE A MORE EFFICIENT METHOD, SUPPOSE YOU ARE TESTING
A DEVICE, THE XY11. SUPPOSE THIS DEVICE CONSISTS OF
A CONTROL MODULE WITH EIGHT UNITS (SUB-DEVICES) ATTACHED TO IT.
THESE UNITS ARE DESCRIBED BY THE OCTAL NUMBERS 0 THROUGH 7. THERE
IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE
Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY
TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

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

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

UNIT 2
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 1<CR>
Q-FACTOR (O) 1 ? 0<CR>

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

UNIT 4
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 3<CR>
Q-FACTOR (O) 0 ? <CR>

UNIT 5
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 4<CR>
Q FACTOR (O) 0 ? <CR>

UNIT 6
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 5<CR>
Q-FACTOR (O) 0 ? <CR>

UNIT 7
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 6<CR>
Q-FACTOR (O) 0 ? 1<CR>

UNIT 8
CSR ADDRESS (O) 160000<CR>
SUB-DEVICE # (O) ? 7<CR>
Q-FACTOR (O) 1 ? <CR>
```

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

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

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

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

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

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

        UNIT 7
        CSR ADDRESS (0) ?  160000<CR>
        SUB-DEVICE # (0) ?  6,7<CR>
        Q-FACTOR (0) 0 ?  1<CR>
```

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

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

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

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

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING A NULL FIELD) TELL THE RUNTIME SERVICES TO REPEAT THE LAST REPLY.

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK) QUESTIONS
3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE "START"
5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THESE DEFAULTS ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SECTION 2.3). THE GENERAL ERROR MESSAGE IS OF THE FORM:

```
NAME  TYPE  NUMBER  ON UNIT NUMBER  TST NUMBER PC:XXXXXX
ERROR MESSAGE

.WHERE; NAME = DIAGNOSTIC NAME
        TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
        NUMBER = ERROR NUMBER
        UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
        TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
        PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL
```

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBR" OR "IXR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 SPECIFIC ERROR MESSAGES

BELOW ARE SAMPLE ERROR MESSAGES. EACH ERROR MESSAGE REPRESENTS DIFFERENT TYPES OF ERRORS DETECTED BY THIS DIAGNOSTIC.

ERROR MESSAGE EXAMPLE 1

THIS ERROR IS INDICATIVE OF AN INCORRECT REGISTER OR STATUS WORD RETURNED TO THE DIAGNOSTIC. THE FIRST PART DEFINES THE TEST FUNCTION AND UNIT THAT FAILED. THE SECOND PART PRIOVIDES THE REGISTER BITS AND THEIR MNEMONICS FOR THE INCORRECT REGISTER OR STATUS WORDS. THE THIRD PART IS THE EXPECTED AND RECEIVED DATA.

```
TST: 016 FIFO EXERCISER TEST
CVTSB 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

THIS ERROR SHOWS A FATAL FUNCTION ERROR FROM THE TAPE DRIVE.IN THIS INSTANCE A UNRECOVERABLE ERROR OCCURED WHICH INDICATES THAT THE CONTROLLER MAY BE DEFECTIVE.

```
CVTSB 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

THIS ERROR SHOWS THAT THE MOTION BIT DID NOT GET SET WHILE DOING A REWING
WITH EXTENDED FEATURES MODE ENABLED.

```
CVTSB 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
```

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE
TOTAL NUMBER OF ERRORS REPORTED SINCE THE DIAGNOSTIC WAS STARTED.
THE "EOP" SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END
OF PASS MESSAGE IS PRINTED. SECTION 2.2 DESCRIBES SWITCHES.

SUCCESSFUL RUN EXAMPLE (PDP-11/23)

```
 DR>STA/FLA:PNT:HOE:UAM

 UNITS (D) ?  1

 UNIT 0

 DEVICE ADDRESS  (O) 172520 ? <CR>

 VECTOR (O) 224 ? <CR>

 CHANGE SW (L) ? N<CR>
```

THE ABOVE COMMAND WILL START THE DIAGNOSTIC. THE COMMAND HAS THREE
SWITCHES ON WHICH ARE "PRINT EACH TEST NBR AS EXECUTED" , "HALT ON
ERROR" AND "RUN IN UNATTENDED MODE".

NOTE:THE UAM FLAG SHOULD BE USED TO PREVENT TEST 10-12 FROM BEING
EXECUTED UNLESS THE OPERATOR WANTS THESE SPECIFIC TEST.

```
TST: 001 INITIALIZE #3 TEST
TST: 002 BASIC WRITE SUBSYSTEM MEMORY TEST
TST: 003 DMA MEMORY ADDRESSING TEST
TST: 004 RAM EXERCISER TEST
TST: 005 FIFO EXERCISER TEST
TST: 006 STATIC TRANSPORT BUS CHECK
TST: 007 TRANSPORT BUS INTERFACE CHECK VIA LOOPBACK TEST
TST: 008 READ/WRITE DATA PARITY CHECK TEST
TST: 009 MISCELLANEOUS LOGIC CHECKS TEST
TST: 010 STAND-ALONE MANUAL INTERVENTION NOT EXECUTED TEST
TST: 011 STAND-ALONE CONFIGURATION TYPEOUT NOT EXECUTED TEST
TST: 012 STAND-ALONE SCOPE LOOPS NOT EXECUTED TEST


0 ERRORS
```

NOTE: THE DIAGNOSTIC WILL RUN CONTINUOUSLY UNLESS A PASS LIMIT HAS BEEN SPECIFIED WITH THE "/PASS:" SWITCH.

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-11/23 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	15	50	35
2	1	6	5
3	1	1	0
4	110	540	430
5	1	10	9
6	10	120	110
7	1	3	2
8	15	15	12
9	17	17	13

THE TIMES REQUIRED TO RUN TESTS 1 THROUGH 9 IN ONE COMMAND:

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

5.0 DEVICE INFORMATION TABLES

WHENEVER THE PROGRAM IS STARTED, VIA THE STA(RT) COMMAND, THE SUPERVISOR REQUESTS THE FOLLOWING P-TABLES PARAMETER CHANGES:

```
CHANGE HW (L) ?

# UNITS (D) ?  <ENTER THE NUMBER OF M7196 CONTROLLERS
                PRESENT TO BE TESTED>

UNIT 0

DEVICE ADDRESS  (0) 172520 ? <ENTER THE ADDRESS OF THE
                              TSBA/TSDB REGISTER>

VECTOR (0) 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.

IN ADDITION, ON A START, RESTART OR CONTINUE THE SUPERVISOR REQUESTS CHANGES TO THE SOFTWARE OPERATING PARAMETERS, AS FOLLOWS:

```
CHANGE SW (L) ?
```

6.0 TEST SUMMARIES

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

TEST DESCRIPTION:

This test verifies that a Hardware Initialize command invoked after a Write Characteristics command sets up the Command, Message and Characteristic image blocks in the controller ram correctly.

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A BSELO SELECT CODE OF 0 (NO-OP) EXECUTES CORRECTLY. IT ALSO VERIFIES THAT A WRITE SUBSYSTEM MEMORY COMMAND WITH A NON-ZERO MODE FIELD IS REJECTED. THE TEST FURTHER VERIFIES MICROPROGRAM COMMAND DECODING AND HANDLING SEQUENCES.

TEST 3: DMA MEMORY ADDRESSING

This test verifies that the controller can properly address and access all available CPU memory (other than that occupied by the diagnostic and diagnostic supervisor code) for both reading (DATI) and writing (DATO). Verified are the LSI-11 Bus drivers for all available address lines. Up to this point only 16 bits have been used for DMA transfers.

****
CAUTION

The LSI BUS drivers for all available address lines are only checked when running on a 11/23B system with more than 128K words of memory!

****

TEST 4: RAM EXERCISER TEST

THIS TEST USES THE READ AND WRITE RAM (BOTH SINGLE AND 256 LOCATIONS) SELECT CODES OF THE WRITE SUBSYSTEM MEMORY COMMAND TO EXERCISE THE CONTROLLER'S RAM MEMORY AND DMA LOGIC

TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B

TEST DESCRIPTION:

This test verifies the Invert Extended Features function can logically invert the Extended features switch and that the internal timers A and B operate correctly.

TEST 6: 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 7: STATIC TRANSPORT BUS INTERFACE TEST

TEST DESCRIPTION:

Write to TSSR register to soft initialize the controller
Do WRITE CHARACTERISTICS to check for Extended Features Switch
If Extended Features Hardware Switch Clear then:
  Do Write Subsystem Write Miscellaneous to Set Extended Features.
Do WRITE CHARACTERISTICS to select reserved unit 7
Do a Write Subsystem READ STATUS
If any transport interface signals are asserted then Print Error

TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST

TEST DESCRIPTION:

This test verifies the controller's Transport Bus drivers, receivers, and signal loopback logic. Note that the Static Transport Bus test must have run correctly for this test to provide meaningful results.

TEST 9: READ/WRITE DATA PARITY TEST

TEST DESCRIPTION:

This test verifies that the Write Data Parity generator and the Read Data Parity checker operate properly. The Transport Bus signal loopback mode is enabled and a Set Wrong parity function is executed. Then various Write Subsystem Memory functions are performed to write data to and from the FIFO in loopback mode. The program then checks to insure a Read Data parity error occurred.
A Reset FIFO is done and the Read Data parity error bit is again tested to insure it cleared.
Finally a Clear wrong parity function is done and it is verified the data word can pass in loopback mode without setting Read Data parity error.

TEST 10: 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 6.
SELECTION CODES AND SUBROUTINES ARE:

CODE	ROUTINE
0	HELP. PRINTS THIS MENU.
1	TURN ON ALL M7196 LED INDICATORS
2	TURN OFF ALL M7196 LED INDICATORS
3	OFFLINE/ONLINE ATTENTION TEST
4	WRITE-PROTECT TEST
5	PRINT EXTENDED TRANSPORT STATUS
6	EXIT (RETURN TO SUPERVISOR)

TEST 11: CONFIGURATION TYPEOUT

THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL
THE CONFIGURATION OF THE M7196 MODULE AND TSV05 SUBSYSTEM. SPECIFICALLY,
THE FOLLOWING INFORMATION IS PRESENTED:

1.0 STATE OF THE EXTENDED FEATURES SWITCH ON THE M7196: ON (EXTENDED
FEATURES ENABLED) OR OFF (EXTENDED FEATURES DISABLED).

2.0 STATE OF THE BUFFERING ENABLE SWITCH ON THE M7196: ON
(BUFFERING ENABLED) OR OFF (BUFFERING DISABLED).

3.0 MICROCODE REVSION LEVEL OF THE M7196.

4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.

5.0 UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED)
OF EACH CONNECTED TRANSPORT. IN ADDITION, THE PROGRAM WILL INDICATE,
FOR EACH ON-LINE TRANSPORT, WHETHER OR NOT IT IS EQUIPPED WITH THE
EXTENDED TAPE STATUS READOUT FEATURE.

TEST 12: SCOPE LOOPS

THIS IS A STANDALONE ROUTINE PROVIDING A NUMBER OF TIGHT "SCOPE LOOPS" USEFUL FOR DEBUGGING BASIC REGISTER ACCESS PROBLEMS WITH THE M7196 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. 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	TSDBX (TSSR HIGH BYTE) WRITE ACCESS (EXTENDED FEATURES SWITCH MUST BE ON TO USE SELECTION CODE 7)
8	EXIT (RETURN TO SUPERVISOR)

FOR SCOPE LOOPS THAT WRITE INTO REGISTERS, THE PROGRAM PROMPTS THE OPERATOR FOR THE DATA TO BE WRITTEN.LIMITS ON THE DATA PATTENS ARE 0-377. TYPING <RETURN> CAUSES AN EXIT FROM THE SCOPE LOOP BACK TO MENU LEVEL.

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

REVISION B - APRIL 1983

MODIFIED THE DIAGNOSTIC TO HANDLE 11/23A'S WITH MORE THAN 256KB OF MEMORY. CHANGED TEST 3 SUBTEST 3 SO IT WON'T TRY TO CREATE NON-EXISTANT MEMORY ADDRESS (NXM).

REVISION C - JUNE 1984

MINOR CHANGES FOR "ORION" CPU
ELIMINATED CPU TYPE IDENTIFICATION MESSAGE.

REVISION D - JULY 1985

CHANGES MADE TO BECOME COMPATIBLE WITH THE XXDP+ V2.1 XM (EXTENDED MONITOR), AND TO ALLOW THE EXTENDED FEATURES TO WORK IN EITHER THE ON OR OFF POSITION.

```
                                        .TITLE  TSV2 - PROGRAM HEADER
                                        .SBTTL  PROGRAM HEADER

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

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

002000                                  POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
002000                                  HEADER  CVTSB,D,0,655.,0
002000                          L$NAME::                ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  126                             .ASCII /V/
002002  124                             .ASCII /T/
002003  123                             .ASCII /S/
002004  102                             .ASCII /B/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                          L$REV::                 ;REVISION LEVEL
002010  104                             .ASCII  /D/
002011                          L$DEPO::                ;0
002011  060                             .ASCII  /0/
002012                          L$UNIT::                ;NUMBER OF UNITS
002012  000000                          .WORD   0
002014                          L$TIML::                ;LONGEST TEST TIME
002014  001217                          .WORD   655.
002016                          L$HPCP::                ;POINTER TO H.W. QUES.
002016  101242                          .WORD   L$HARD
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  101374                          .WORD   L$SOFT
002022                          L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002156                          .WORD   L$HW
002024                          L$SPTP::                ;PTR. TO S.W. PTABLE
002024  002166                          .WORD   L$SW
002026                          L$LADP::                ;DIAG. END ADDRESS
002026  102004                          .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  002124                          .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  003402                  .WORD   L$DVTYP
002062                  L$REPP::                ;PTR. TO REPORT CODE
002062  022214                  .WORD   L$RPT
002064                  L$EXP4::
002064  C00000                  .WORD   0
002066                  L$EXP5::
002066  000000                  .WORD   0
002070                  L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  021702                  .WORD   L$AU
002072                  L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022000                  .WORD   L$DU
002074                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                  .WORD   0
002076                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  003410                  .WORD   L$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  021106                  .WORD   L$INIT
002106                  L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  022166                  .WORD   L$CLEAN
002110                  L$ACP::                 ;PTR. TO AUTO CODE
002110  022106                  .WORD   L$AUTO
002112                  L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021076                  .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
```

DISPATCH TABLE

```
    959                                         .SBTTL  DISPATCH TABLE
    960
    961                                 ;++
    962                                 ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
    963                                 ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
    964                                 ;--
    965
    966 002122                                  DISPATCH 12
        002122  000014                          .WORD   12
        002124                          L$DISPATCH::
        002124  022776                          .WORD   T1
        002126  023756                          .WORD   T2
        002130  025750                          .WORD   T3
        002132  031344                          .WORD   T4
        002134  034134                          .WORD   T5
        002136  037726                          .WORD   T6
        002140  050040                          .WORD   T7
        002142  051320                          .WORD   T8
        002144  062146                          .WORD   T9
        002146  066216                          .WORD   T10
        002150  074054                          .WORD   T11
        002152  077564                          .WORD   T12
```

DEFAULT HARDWARE P-TABLE

```
                                        .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.
                                ;--
002154                                  BGNHW   DFPTBL          ;DEFAULT HARD-P-TABLE
002154  000003                          .WORD   L10000-L$HW/2
002156                          L$HW::
002156                          DFPTBL::

002156  172520                          .WORD   172520          ; 1ST (OF 2) REGISTERS.
002160  000224                          .WORD   224             ; INTERRUPT VECTOR
002162  000200                          .WORD   PRI04           ; INTERRUPT PRIORITY.
002164                                  ENDHW
002164                          L10000:
```

SOFTWARE P-TABLE

```
                                        .SBTTL  SOFTWARE P-TABLE

                                ;++
                                ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
                                ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                                ;--
002164                                  BGNSW   SFPTBL
002164  000004                         .WORD   L10001-L$SW/2
002166                          L$SW::
002166                          SFPTBL::

002166  000000                  TRANSTST::      .WORD   0       ; ENABLE TEST OF TRANSPORT(S) IF =1
002170  000000                  NOITS::         .WORD   0       ; INHIBIT ITERATION OPTION.
                                                                ; ... 0 = ITERATE.
                                                                ; ...NZ = INHIBIT ITERATE.
002172  000017                  LERRMAX::       .WORD   15.     ; LOCAL (PER TEST) ERROR LIMIT
002174  000310                  GERRMAX::       .WORD   200.    ; GLOBAL (PER UNIT) ERROR LIMIT
002176                                  ENDSW
002176                          L10001:

                                        ENDMOD
```

SOFTWARE P-TABLE

```
1008                                    .TITLE  TSV3 - GLOBAL AREAS
1009                                    .SBTTL  GLOBAL EQUATES SECTION
1014
1020
1021 002176                             BGNMOD  TSV3
     002176                         TSV3::
1022
1023                                    .SBTTL  GLOBAL EQUATES SECTION
1024
1025                                    ;++
1026                                    ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
1027                                    ; ARE USED IN MORE THAN ONE TEST.
1028                                    ;--
1029
1033 002176                             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
```

GLOBAL EQUATES SECTION

```
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
000140                  PRI03== 140
000100                  PRI02== 100
000040                  PRI01== 40
000000                  PRI00== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==         4
000010                  LOT==        10
000020                  ADR==        20
000040                  IDU==        40
000100                  ISR==       100
000200                  UAM==       200
000400                  BOE==       400
001000                  PNT==      1000
002000                  PRI==      2000
004000                  IXE==      4000
010000                  IBE==     10000
020000                  IER==     20000
040000                  LOE==     40000
100000                  HOE==    100000
1034
1035 002176                       KT11                    ;DEFINE MEMORY MANAGEMENT REGISTERS
                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS
                        ;*KT11 VECTOR ADDRESS
000250                  MMVEC=  250
                        ;*KT11 STATUS REGISTER ADDRESSES
177572                  SR0=    177572
177574                  SR1=    177574
177576                  SR2=    177576
172516                  SR3=    172516
                        .IF NB
                        ;*USER "I" PAGE DESCRIPTOR REGISTERS
                        UIPDR0= 177600
                        UIPDR1= 177602
                        UIPDR2= 177604
                        UIPDR3= 177606
                        UIPDR4= 177610
                        UIPDR5= 177612
                        UIPDR6= 177614
                        UIPDR7= 177616
                        .IF NB
                        ;*USER "D" PAGE DESCRIPTOR 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 DESCPIPTOR 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
```

MEMORY MANAGEMENT DEFINITIONS

```
                        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  TSV05 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
177520                  BDVPCR==        177520          ; BDV11 PAGE CONTROL REGISTER

                        ;+
                        ;BIT DEFINITIONS FOR TSSR REGISTER
                        ;-

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

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

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

TSV05 REGISTER AND PACKET DEFINITIONS

```
1092                                    ;+
1093                                    ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
1094                                    ;(XST1)
1095                                    ;-
1096        100000                      X1.DLT  = BIT15         ;DATA LATE
1097        040000                      X1.SPARE= BIT14         ;NOT USED
1098        020000                      X1.COR  = BIT13         ;CORRECTABLE DATA ERROR
1099        017375                      X1.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
1100        000400                      X1.RBP  = BIT8          ;READ BUS PARITY ERROR
1101        000002                      X1.UNC  = BIT1          ;UNCORRECTABLE DATA OR HARD ERROR
1102
1103                                    ;+
1104                                    ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
1105                                    ;(XST2)
1106                                    ;-
1107        100000                      X2.OPM  = BIT15         ;OPERATION IN PROGRESS (TAPE MOVING)
1108        040000                      X2.RCE  = BIT14         ;RAM CHECKSUM ERROR
1109        035400                      X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8   ;NOT USED BY TSV05 (ALWAYS=0)
1110        002000                      X2.WCF  = BIT10         ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
1111        000200                      X2.EXTF = BIT7          ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
1112        000100                      X2.BUFE = BIT6          ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
1113        000077                      X2.REV  = 000077        ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
1114        000007                      X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
1115
1116                                    ;+
1117                                    ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
1118                                    ;(XST3)
1119                                    ;-
1120        177400                      X3.MDE  = 177400        ;MICRO-DIAGNOSTIC ERROR CODE
1121        000200                      X3.SPARE= BIT7          ;NOT USED BY TSV05
1122        000100                      X3.OPI  = BIT6          ;OPERATION INCOMPLETE
1123        000040                      X3.REV  = BIT5          ;REVERSE
1124        000020                      X3.TRF  = BIT4          ;TRANSPORT RESPONSE FAILURE
1125        000010                      X3.DCK  = BIT3          ;DENSITY CHECK
1126        000006                      X3.MBZ  =BIT2+BIT1      ;NOT USED ALWAYS 0
1127        000001                      X3.RIB  = BIT0          ;REVERSE INTO BOT
1128
1129                                    ;+
1130                                    ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
1131                                    ;(XST4)
1132                                    ;-
1133        100000                      X4.HSP  = BIT15         ;HIGH SPEED
1134        040000                      X4.RCE  = BIT14         ;RETRY COUNT EXCEEDED
1135        020000                      X4.TSM  = BIT13         ;TRANSPORT SPECIAL MODE
1136        017400                      X4.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT8   ;NOT USED ALWAYS 0
1137        000377                      X4.WRC  = 000377        ;WRITE RETRY COUNT FIELD
1138
1139                                    ;+
1140                                    ;
1141                                    ;TSSR TERMINATION CODES (BIT 0-2)
1142                                    ;
1143                                    ;-
1144
1145        000006                              TSREJ=  3+2             ;COMMAND REJECTED
1146        000006                              UNREC=  6               ;UNRECOVERABLE ERROR
```

```
1148                                    ;*
1149                                    ;
1150                                    ;DEVICE REGISTER OFFSETS
1151                                    ;
1152                                    ;-
1153
1154    000000                          TSBA== 0
1155    000000                          TSDB== 0                ;TSDB/TSBA REGISTER
1156    000001                          TSBAH== 1
1157    000001                          TSDBH== 1               ;TSDB/TSBA REGISTER HIGH BYTE
1158    000002                          TSSR== 2                ;TSSR REGISTER
1159    000003                          TSSRH== 3               ;TSSR REGISTER HIGH BYTE
1160
1161                                    ;*
1162                                    ; TSDB ADDRESS BIT DEFINITIONS
1163                                    ;-
1164    000003                          A1716   = BIT1+BIT0     ;ADDRESS BITS 17:16 ARE IN 1:0
1165
1166                                    ;*
1167                                    ; COMMAND DEFINITIONS
1168                                    ;-
1169    000017                          P.GETSTAT       = 17    ;GET STATUS
1170    000013                          P.INIT          = 13    ;INITIALIZE
1171    000012                          P.CONTROL       = 12    ;CONTROL COMMANDS
1172    000011                          P.FORMAT        = 11    ;FORMAT
1173    000010                          P.POSITION      = 10    ;POSITION
1174    000006                          P.WRTSUB        = 6     ;SUBSYSTEM WRITE
1175    000005                          P.WRITE         = 5     ;WRITE
1176    000004                          P.WRTCHAR       = 4     ;WRITE CHARACTERISTICS
1177    000001                          P.READ          = 1     ;READ
1178
1179                                    ;*
1180                                    ; COMMAND PACKET HEADER WOPD BIT DEFINITIONS
1181                                    ;-
1182    100000                                  P.ACK   = BIT15         ;BUFFER AVAIL FOR CONTROLLER
1183    040000                                  P.CVC   = BIT14         ;CLEAR VOLUME CHECK
1184    020000                                  P.OPP   = BIT13         ;REVERSE SEQUENCE OF DATA BITS
1185    010000                                  P.SWB   = BIT12         ;SWAP BYTES IN MEMORY
1186    007400                                  P.MODE  = BIT11!BIT10!BIT9!BIT8 ;EXTENDED COMMAND MODE FIELD
1187    000200                                  P.IE    = BIT7          ;INTERRUPT ENABLE
1188    000140                                  P.FMT= BIT6!BIT5        ;PACKET HEADER TYPE (ALWAYS=0)
1189    000037                                  P.CMD   = 37            ;MAJOR COMMAND FIELD
1190
1191                                    ;*
1192                                    ; CONTROL COMMAND MODE CODES
1193                                    ;-
1193    000000                                  PC.RELEASE      = 0*256.        ;RELEASE BUFFER
1194    000400                                  PC.REWIND       = 1*256.        ;REWIND
1195    001000                                  PC.NOOP         = 2*256.        ;NO-OP
1196    002000                                  PC.IEREW        = 4*256.        ;REWIND IMMEDIATE INTERRUPT
1197    002400                                  PC.ERASE        = 5*256.        ;SECURITY ERASE
```

TSV05 REGISTER AND PACKET DEFINITIONS

```
1199                                ;*
1200                                ; CONTROLLER RAM DEFINITIONS
1201                                ;-
1202    000167                      RMCHBEG = 167           ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
1203    000200                      RMCHEND = 200           ;CHARACTERISTICS IO DATA END RAM ADDRESS
1204    000201                      RMPKTBEG= 201           ;COMMAND PACKET BEGIN RAM ADDRESS
1205    000210                      RMPKTEND= 210           ;COMMAND PACKET END RAM ADDRESS
1206    000215                      RMMSGBEG= 215           ;MESSAGE BUFFER BEGIN RAM ADDRESS
1207    000234                      RMMSGEND= 234           ;MESSAGE BUFFER END RAM ADDRESS
1208                                ;*
1209                                ;
1210                                ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
1211                                ;
1212                                ;-
1213
1214    000006                      XST0==  6               ;EXTENDED STATUS REGISTER 0 (WORD 4)
1215    000010                      XST1==  8.              ;EXTENDED STATUS REGISTER 1 (WORD 5)
1216    000012                      XST2==  10.             ;EXTENDED STATUS REGISTER 2 (WORD 6)
1217    C00014                      XST3==  12.             ;EXTENDED STATUS REGISTER 3 (WORD 7)
1218    000016                      XST4==  14.             ;EXTENDED STATUS REGISTER 4 (WORD 8)
1219
1220                                ;*
1221                                ;
1222                                ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
1223                                ;
1224                                ;-
1225
1226    000002                              PKLOW   = 2         ;LOW ORDER CHARACTERISTIC DATA POINTER
1227    000004                              PKHI    = 4         ;HIGH ORDER CHARACTERISTIC DATA POINTER
1228    000006                              PKBCNT  = 6         ;NUMBER OF BYTES IN DATA PACKET
1229
1230    000010                              EXBCNT=10           ;NUMBER OF BYTES IN EXTENDED DATA PACKET
1231
1232                                ;*
1233                                ;DATA PACKET OFFSETS FOR WRITE SUBSYTEM COMMAND
1234                                ;-
1235    000000                              BSEL0   = 0         ;BYTE 0
1236    000001                              BSEL1   = 1         ;BYTE 1
1237    000002                              SEL2    = 2         ;WORD 2
1238    000004                              SELDATA = 4         ;WORD 3
```

```
1240                                    ;+
1241                                    ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
1242                                    ;-
1243            000000                  PW.NOP          = 0             ;NO-OP
1244            000001                  PW.RDRAM        = 1             ;READ RAM
1245            000002                  PW.WTRAM        = 2             ;WRITE RAM
1246            000003                  PW.RFIFO        = 3             ;READ FIFO
1247            000004                  PW.WFIFO        = 4             ;WRITE FIFO
1248            000005                  PW.RDSTAT       = 5             ;READ STATUS
1249            000006                  PW.WCTL         = 6             ;WRITE TAPE CONTROL
1250            000007                  PW.WFMT         = 7             ;WRITE TAPE FORMAT
1251            000010                  PW.WMISC        = 10            ;WRITE MISCELLANEOUS
1252            000011                  PW.WNPR         = 11            ;WRITE NPR CONTROL
1253            000020                  PW.D22          = 20            ;DO MICROTEST 22
1254            000021                  PW.D11          = 21            ;DO MICROTEST 11
1255            000022                  PW.D13          = 22            ;DO MICROTEST 13
1256            000023                  PW.ND1311       = 23            ;DISABLE MICROTEST 11 AND 13
1257            000024                  PW.RDEXT        = 24            ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS
1258
1259                                    ;+
1260                                    ;BSEL1 CODES FOR WRITE TAPE CONTROL
1261                                    ;-
1262            000200                  WC.IFAD         = BIT7          ;IFAD - FORMATTER ADDRESS
1263            000100                  WC.IOTAD        = BIT6          ;ITAD0  - TRANSPORT ADDRESS BIT 0
1264            000040                  WC.I1TAD        = BIT5          ;ITAD1  - TRANSPORT ADDRESS BIT 1
1265            000020                  WC.I5RESV       = BIT4          ;IRESV5 - RESERVED #5
1266            000010                  WC.IREW         = BIT3          ;IREW   - REWIND
1267            000004                  WC.IRWU         = BIT2          ;IRWU   - REWIND AND UNLOAD
1268            000002                  WC.IFEN         = BIT1          ;IFEN   - FORMATTER ENABLE
1269            000001                  WC.IGO          = BIT0          ;GO
1270
1271                                    ;+
1272                                    ;BSEL1 CODES FOR WRITE FORMAT
1273                                    ;-
1274            000200                  WF.IHISP        = BIT7          ;IHISP  - HIGH SPEED
1275            000100                  WF.IWRT         = BIT6          ;IWRT   - WRITE
1276            000040                  WF.IREV         = BIT5          ;IREV   - REVERSE
1277            000020                  WF.IWFM         = BIT4          ;IWFM   - WRITE FILE MARK
1278            000010                  WF.IEDIT        = BIT3          ;IEDIT  - EDIT
1279            000004                  WF.IERASE       = BIT2          ;IERASE - ERASE
1280            000002                  WF.I3RESV       = BIT1          ;IRESV3 - RESERVED #3
1281            000001                  WF.I4RESV       = BIT0          ;IRESV4 - RESERVED #4
1282
1283                                    ;+
1284                                    ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
1285                                    ;-
1286            000200                  MS.EXT          = BIT7          ;INVERT SENSE OF EXTENDED FEATURES SWITCH
1287            000020                  MS.RSFIFO       = BIT4          ;RESET FIFO AND INPUT PARITY ERRORR
1288            000010                  MS.RSTAPE       = BIT3          ;RESET TAPE STATUS IN 2 FLIP-FLOPS
1289            000006                  MS.ATTN         = BIT2!BIT1     ;ATTENTION TRIGGER FIELD
1290            000001                  MS.RSD          = BIT0          ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
```

```
                                ;+
                                ; MS.ATTN SUBCODES
                                ;-
000000                                  MSA.NOP = 0*2           ;NO-OP (NOTHING TRIGGERED)
000002                                  MSA.VOL = 1*2           ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
000004                                  MSA.NRAM= 2*2           ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
000006                                  MSA.FRAM= 3*2           ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)

                                ;+
                                ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
                                ;-
000200                          NP.IR           = BIT7          ;INTERRUPT REQUEST (0-1 TRANSITION)
000100                          NP.OUT          = BIT6          ;TAPE DATA DIRECTION OUT (0= IN)
000040                          NP.LOOP         = BIT5          ;ENABLE TRANSPORT LOOPBACK
000020                          NP.WRP          = BIT4          ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)

                                ;+
                                ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
                                ;-

C00200                          S2.DIM          = BIT7          ;WORD #9 BYTE 2 DATA IN MISS
000100                          S2.ILW          = BIT6          ;                ILW H
000040                          S2.OUTRDY       = BIT5          ;                OUT RDY H
000020                          S2.INRDY        = BIT4          ;                IN RDY H
000010                          S2.ATIMR        = BIT3          ;                TIMER A FLAG H
000004                          S2.BTIMR        = BIT2          ;                TIMER B FLAG H
000003                          S2.UNDEF        = BIT1+BIT0     ;(UNDEFINED)
100000                          S1.PARIN        = BIT15         ;WORD #8 BYTE 1 PARIN H
040000                          S1.I2RESV       = BIT14         ;                IRESV2
020000                          S1.I1RESV       = BIT13         ;                IRESV1
010000                          S1.IEOT         = BIT12         ;                IEOT L
004000                          S1.IIDENT       = BIT11         ;                IIDENT H
002000                          S1.ICER         = BIT10         ;                ICER H
001000                          S1.IFMK         = BIT9          ;                IFMK H
000400                          S1.IHER         = BIT8          ;                IHER H
000200                          S0.ISPEED       = BIT7          ;WORD #8 BYTE 0 ISPEED H
000100                          S0.IRDY         = BIT6          ;                IRDY L
000040                          S0.IONL         = BIT5          ;                IONL L
000020                          S0.ILDP         = BIT4          ;                ILDP L
000010                          S0.IDBY         = BIT3          ;                IDBY L
000004                          S0.IRWD         = BIT2          ;                IRWD L
000002                          S0.IFBY         = BIT1          ;                IFBY L
000001                          S0.IFPT         = BIT0          ;                IFPT L
```

SPECIAL MACROS AND OPDEFS.

```
1334                                    .SBTTL  SPECIAL MACROS AND OPDEFS.
1335
1336                                ;+
1337                                ;SAVE GENERAL REGS 1 TO 5
1338                                ;-
1339
1340                                    .MACRO  SAVREG
1341                                    JSR     R5,REGSAV
1342                                    .ENDM
1343
1344                                ;+
1345                                ; MACRO TO FORCE AN ERROR
1346                                ;-
1347                                .MACRO  FORCERROR       TAG,NOTSSR
1348                                .NLIST
1349                                .IIF NDF LISTALL, .NLIST
1350                                .LIST
1351                                .IF B NOTSSR
1352                                    MOV     TSSR(R5),R1     ;READ TSSR
1353                                .ENDC
1354                                    MOV     FORCER,FORCER   ;IS FORCER SET? (LEAVE C BIT ALONE)
1355                                    BNE     TAG             ;BR IF YES
1356                                .NLIST
1357                                .IIF NDF LISTALL, .LIST
1358                                .LIST
1359                                .ENDM
1360
1361                                ;+
1362                                ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
1363                                ;       WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
1364                                ;       SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
1365                                ;       FORCER TO 177777
1366                                ;       TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
1367                                ;-
1368                                .MACRO  FORCEXIT        TAG
1369                                .NLIST
1370                                .IIF NDF LISTALL, .NLIST
1371                                .LIST
1372                                    MOV     FORCER,FORCER   ;IS FORCER NEGATIVE?
1373                                    BMI     TAG             ;BR IF YES
1374                                .NLIST
1375                                .IIF NDF LISTALL, .LIST
1376                                .LIST
1377                                .ENDM
1378                                ;+
1379                                ; MACRO TO INCREMENT ERROR COUNTS
1380                                ;-
1381                                .MACRO  NEXT.ERRNO
1382                                .NLIST
1383                                ;;;.IIF NDF LISTALL, .NLIST
1384                                    ERRNO=ERRNO+1
1385                                ;;;.IIF NDF LISTALL, .LIST
1386                                .LIST
1387                                .ENDM
```

SPECIAL MACROS AND OPDEFS.

```
                                ;+
                                ;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.
                                ;

002176  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.
```

GLOBAL DATA SECTION

```
                                        .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.
                                ;
002200  000000  EPRTSW::        .WORD   0       ;PRINT SWITCH
002202  000000  UNITN::         .WORD   0       ;UNIT # UNDER TEST.
002204  000000  QVP::           .WORD   0       ;QUICK VERIFY FLAG.
002206  000000  CSRADDR::       .WORD   0       ;ADDRESS OF CSR FOR CURRENT DEVICE
002210  000224  IVEC::          .WORD   224     ;INTERRUPT VECTOR
002212  000200  IPRI::          .WORD   PRI04   ;INTERRUPT PRIORITY.
002214  000000  TSTCNT::        .WORD   0       ;NUMBER OF TESTS RUN IN THIS PASS
002216  000000  LOOPCNT::       .WORD   0       ;REMAINING    RATION COUNT FOR TEST
002220  000000  DEVCNT::        .WORD   0       ;NUMBER OF DEVICE UNDER TEST
002222  000000  FATFLG::        .WORD   0       ;SET IF FATAL ERROR IS DETECTED IN TEST
002224  000000  INTRECV::       .WORD   0       ;SET IF TAPE INTERRUPT WAS RECEIVED
002226  000000  EXTFEA::        .WORD   0       ;EXTENDED FEATURES SOFTWARE SW 0=OFF;1=ON
002230  000000  BENBSW::        .WORD   0       ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
002232  000000  EXPD::          .WORD   0       ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
002234  000000  RECV::          .WORD   0       ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
002236  000000  ERRHI::         .WORD   0       ;HIGH ADDRESS MEMORY ERROR
002240  000000  ERRLO::         .WORD   0       ;LOW ADDRESS MEMORY ERROR
002242          RAMDATA::       .BLKW   16.     ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
002302  000000  RAMSIZ::        .WORD   0       ;RAM DATA SIZE FOR PRAMPKT ROUTINE
002304  000000  RCVHIADD::      .WORD   0       ;RECEIVED BUFFER HIGH ADDRESS
002306  000000  RCVLOADD::      .WORD   0       ;RECEIVED BUFFER LOW ADDRESS
002310  000000  COUNT::         .WORD   0       ;TEST COUNT PATTERN
002312  000000  DATA::          .WORD   0       ;TEST DATA
002314  000000  TSTFLAG::       .WORD   0       ;TEST FLAG WORD
002316  000000  TSTPTR::        .WORD   0       ;TSTBLK POINTER
002320  000000  PRMNO::         .WORD   0       ;PRINT ROUTINE TEMP
002322          EXPMSG::        .BLKB   100.    ;EXPECTED MESSAGE BUFFER DATA
002466          RECMSG::        .BLKB   100.    ;RECEIVED MESSAGE BUFFER DATA
002632          TMPBFR::        .BLKB   80.     ;TEMPORARY STORAGE FOR PRINT
```

```
                        .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
                ;
                ;-

002752          TSTBLK::
002752  000000          .WORD   0               ;ALL ZEROS
002754  177777          .WORD   177777          ;ALL ONES
002756  000001          .WORD   BIT0            ;DATA FOR WALKING ONES
002760  000002          .WORD   BIT1
002762  000004          .WORD   BIT2
002764  000010          .WORD   BIT3
002766  000020          .WORD   BIT4
002770  000040          .WORD   BIT5
002772  000100          .WORD   BIT6
002774  000200          .WORD   BIT7
002776  000400          .WORD   BIT8
003000  001000          .WORD   BIT9
003002  002000          .WORD   BIT10
003004  004000          .WORD   BIT11
003006  010000          .WORD   BIT12
003010  020000          .WORD   BIT13
003012  040000          .WORD   BIT14
003014  100000          .WORD   BIT15
003016  177776          .WORD   ^CBIT0          ;DATA FOR WALKING ZEROS
003020  177775          .WORD   ^CBIT1
003022  177773          .WORD   ^CBIT2
003024  177767          .WORD   ^CBIT3
003026  177757          .WORD   ^CBIT4
003030  177737          .WORD   ^CBIT5
003032  177677          .WORD   ^CBIT6
003034  177577          .WORD   ^CBIT7
003036  177377          .WORD   ^CBIT8
003040  176777          .WORD   ^CBIT9
003042  175777          .WORD   ^CBIT10
003044  173777          .WORD   ^CBIT11
003046  167777          .WORD   ^CBIT12
003050  157777          .WORD   ^CBIT13
003052  137777          .WORD   ^CBIT14
003054  077777          .WORD   ^CBIT15
003056  125252          .WORD   125252          ;ALTERNATING ONES, ZEROS
003060  052525          .WORD   052525          ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
003062          TBLEND==.
```

GLOBAL ENVIRONMENT STORAGE

```
                                        .SBTTL  GLOBAL ENVIRONMENT STORAGE
                                ;
                                ;STORAGE FOR DEVICE REGISTERS
                                ;
003062  000000  100000  000000  DUMMY:  0,100000,0,0     ;DUMMY DEVICE REGISTERS...
003072  000000  000000  000000          0,0,0,0,0,0,0,0
                                                        ;...FOR MULTI-UNIT CHECKOUT.

003112  000000                  DUFLG::         .WORD   0               ;"DROPPED UNIT" FLAG.
                                                                        ;INHIBITS CODE IN "CLEAN-UP".
003114  000000                  NODEV::         .WORD   0               ;FLAG TO SAY NO DEVICE.

003116  000000                  TEMP1::         .WORD   0               ;SOME TEMP LOCATIONS.
003120  000000                  TEMP2::         .WORD   0
003122  000000                  XXCOMM::        .WORD   0               ;XXDP+ COMM BLOCK POINTER.
003124  000000                  FREE::          .WORD   0               ;1ST FREE MEMORY ADDRESS...
003126  000000                  FRESIZ::        .WORD   0               ;...AND SIZE (IN WORDS).
003130  000000                  FREEHI: .WORD   0               ;LAST WORD IN FREE SPACE
003132  000000                  KTFLG::         .WORD   0               ;KT11, MEM AVAIL FLAG -
                                                                        ;- .WORD        0 = <24K OR NO KT -
                                                                        ;- NZ = >24K AND KT.
003134  000000                  KTENABLE::      .WORD   0               ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
003136  000000                  NXMFLG::        .WORD   0               ;SET IF WE CAN TEST CLEARED OTHERWISE
003140  000000                  NXMLO::         .WORD   0               ;NXM LO ADDRESS BITS
003142  000000                  NXMHI::         .WORD   0               ;NXM HI ADDRESS BITS FOR DAL'S 16-21
003144  000000                  T23A::          .WORD   0               ;11/23A FLAG
003146  000000                  T238::          .WORD   0               ;11/238 FLAG
003150  000000                  T38FLG::        .WORD   0               ;TEST 38 FLAG +0
003152  002000                  PST32W::        .WORD   2000            ;32W BLOCK ADDRESS FOR 32K START
003154  000000                  SIFLAG::        .WORD   0               ;
003156  000000                  BADDAT::        .WORD   0               ;ACTUAL DATA
003160  000000                  GDDAT::         .WORD   0               ;EXPECTED DATA
003162  000000                  LOOPFL::        .WORD   0
003164                          CTAB::                                  ;CONFIGURATION TABLES.
003164  000000                  CTABM::         .WORD   0               ;CONFIG WORK.
003166  000000                                  .WORD   0
003170  000000                                  .WORD   0
003172  000000                                  .WORD   0
003174  177777                                  .WORD   -1              ;END OF MEM TABLE.
003176                          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
                                ;
003176                          ERTABL:         .BLKW   64.
003376  000000                  ERTABE:         .WORD   0

003400  000000  SKIPT:  .WORD   0               ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST
```

GLOBAL TEXT MESSAGES

```
                                        .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
                                ;-
003402                                  DEVTYP  <TSV05>
003402                          L$DVTYP::
003402  124     123     126             .ASCIZ  /TSV05/
                                        .EVEN

                                ;+
                                ;TEST DESCRIPTION
                                ;-
003410                                  DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****>
003410                          L$DESC::
003410  052     052     052             .ASCIZ  /**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****/
                                        .EVEN


                                ;+
                                ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
                                ;-
003502  003542  003545  003551  TSSRBIT::       .WORD   1$,2$,3$,4$,5$,6$,7$,8$
003522  003603  003607  003613                  .WORD   9$,10$,11$,12$,13$,14$,15$,16$
003542    123     103     000   1$:     .ASCIZ  'SC'
003545    102     111     105   2$:     .ASCIZ  'BIE'
003551    123     103     105   3$:     .ASCIZ  'SCE'
003555    122     115     122   4$:     .ASCIZ  'RMR'
003561    116     130     115   5$:     .ASCIZ  'NXM'
003565    116     102     101   6$:     .ASCIZ  'NBA'
003571    102     111     124   7$:     .ASCIZ  'BIT9'
003576    102     111     124   8$:     .ASCIZ  'BIT8'
003603    123     123     122   9$:     .ASCIZ  'SSR'
003607    117     106     114   10$:    .ASCIZ  'OFL'
003613    102     111     124   11$:    .ASCIZ  'BIT5'
003620    102     111     124   12$:    .ASCIZ  'BIT4'
003625    102     111     124   13$:    .ASCIZ  'BIT3'
003632    102     111     124   14$:    .ASCIZ  'BIT2'
003637    102     111     124   15$:    .ASCIZ  'BIT1'
003644    102     111     124   16$:    .ASCIZ  'BIT0'
                                .EVEN
003652    124     123     123   SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
003705    124     123     123   SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
003740    040     040     116   NXR:    .ASCIZ  /  NON-EXISTANT DEVICE REGISTER/
003777    045     101     040   NXRX:   .ASCIZ  /%A  ADDRESS: %O6/
004020    045     101     040   TSSX:   .ASCII  /%A  TSBA,TSSR EXP'D: %O6%A,%O6%N/
004060    045     101     040           .ASCIZ  /%A  TSBA,TSSR REC'D: %O6%A,%O6/
004117    045     116     045   FUSI:   .ASCII  /%N%A/
004123    040     040     125   USI:    .ASCIZ  /  UNEXPECTED INTERRUPT/
004152    040     040     111   NSI:    .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
004215    045     116     045   FNOINTR:        .ASCII  /%N%A/
004221    040     040     116   NOINTR: .ASCIZ  /  NO INTERRUPT WAS GENERATED'
004256    040     040     111   IFAULT: .ASCIZ  /  INTERRUPT FAULT/
```

GLOBAL TEXT MESSAGES

```
1633 004300      045      101      040  INTX:    .ASCIZ  /%A  CPU PC: %O6%A  TSBA: %O6/
1634 004335      040      040      042  NOINIT:  .ASCIZ  /  "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
1635 004407      040      040      042  NSINIT:  .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
1636 004457      040      040      042  BRINIT:  .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/
1637 004527      000               000  NUL:     .ASCIZ  //
1638 004530      045      116      000  NULCR:   .ASCIZ  /%N/
1639 004533      045      101      040  EXPGOT:  .ASCIZ  /%A EXP'D: %O6%A, REC'D: %O6/
1640 004567      045      116      045  EXPGT2:  .ASCIZ  /%N%A EXP'D: %O6%A, %O6%N%A REC'D: %O%A, %O6/
1641 004643      045      101      040  DUAD12:  .ASCIZ  /%A REG(W) WRITTEN TO: %O6%A REG(R) READ; EXP'D: %O6%A, REC'D: %O6/
1642 004745      122      101      115  PKTRAM::         .ASCIZ  'RAM Contents Do Not Match Packet Sent'
1643 005013      040      040      103  SCME:    .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
1644 005056      127      122      111  WRTMSG:  .ASCIZ  'WRITE CHARACTERISTICS Failed'
1645 005113      124      123      123  WRTERR:  .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
1646 005206      124      123      123  RDERR:   .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
1647 005300      106      101      124  SCHERR:  .ASCIZ  'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
1648 005372      105      122      122  RETERR:  .ASCIZ  'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
1649 005460      045      116      045  NOMEM:   .ASCIZ  '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. ******%N'
1650 005554      045      116      045  M8186:   .ASCIZ  '%N%A ****************** 11/23A SYSTEM ******************%N'
1651 005645      045      116      045  M8189:   .ASCIZ  '%N%A ****************** 11/23B SYSTEM ******************%N'
1652                                               .EVEN
1653                                               .SBTTL  GLOBAL ERROR REPORT SECTION
1654
1655                                     ;++
1656                                     ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
1657                                     ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
1658                                     ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
1659                                     ;--
1660 005736                                        BGNMSG  NXRERR          ;NON-EXISTANT DEVICE REGISTER.
     005736                             NXRERR::
1661 005736                                        PRINTX  #NXRX,NODEV     ;NODEV = NEXM ADDRESS.
     005736  013746  003114                        MOV     NODEV,-(SP)
     005742  012746  003777                        MOV     #NXRX,-(SP)
     005746  012746  000002                        MOV     #2,-(SP)
     005752  010600                                MOV     SP,R0
     005754  104415                                TRAP    C$PNTX
     005756  062706  000006                        ADD     #6,SP
1662 005762  004737  005770                        JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
1663 005766                                        ENDMSG
     005766                             L10002:
     005766  104423                                TRAP    C$MSG
1664                                     ;
1665                                     ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
1666                                     ; TO ANY OF THE ABOVE ERROR SIGNATURES.
1667                                     ;
1668 005770  005727                      EXTEND: TST     (PC)+
1669 005772  000000                      EXTA:   0                       ; 0 = NO EXTENSION.
1670 005774  001402                              BEQ     1$
1671 005776  004777  177770                      JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
1672 006002                              1$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
     006002  012746  004530                      MOV     #NULCR,-(SP)
     006006  012746  000001                      MOV     #1,-(SP)
     006012  010600                              MOV     SP,R0
     006014  104415                              TRAP    C$PNTX
     006016  062706  000004                      ADD     #4,SP
1673 006022  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
                                ;
                                ;-

006024                          PRITSSR:
006024                                  SAVREG                          ;SAVE GENERAL REGISTERS
006030  010104                          MOV     R1,R4                   ;SAVE THE TSSR CONTENTS
006032                                  PRINTB  #TSSRFOR,R4             ;PRINT THE CONTENTS OF TSSR
006032  010446                          MOV     R4,-(SP)
006034  012746  006415                  MOV     #TSSRFOR,-(SP)
006040  012746  000002                  MOV     #2,-(SP)
006044  010600                          MOV     SP,R0
006046  104414                          TRAP    C$PNTB
006050  062706  000006                  ADD     #6,SP
006054  010400                          MOV     R4,R0                   ;GET TSSR BACK FOR CHKAMB
006056  004737  016044                  JSR     PC,CHKAMB               ;ARE CONTENTS AMBIGUOUS ?
006062  103410                          BCS     5$                      ;BRANCH IF NOT
006064                                  PRINTX  #AMBTSSR                ;SHOW CONTENTS ARE AMBIGUOUS
006064  012746  006635                  MOV     #AMBTSSR,-(SP)
006070  012746  000001                  MOV     #1,-(SP)
006074  010600                          MOV     SP,R0
006076  104415                          TRAP    C$PNTX
006100  062706  000004                  ADD     #4,SP
006104  010403                  5$:     MOV     R4,R3                   ;CONTENTS OF TSSR
006106  042703  001476                  BIC     #HIADDR!FATERR!TERCLS,R3        ;CLEAR ALL MULTIPLE BIT FIELDS
006112  001434                          BEQ     20$                     ;NO BITS ARE SET
006114  012702  002632                  MOV     #TMPBFR,R2              ;TEMPORARY ASCII BUFFER
006120  012701  003502                  MOV     #TSSRBIT,R1             ;ASCII EQUIVALENT OF BITS
006124  005703                  10$:    TST     R3                      ;REMAINING BITS TO CONVERT
006126  001413                          BEQ     15$                     ;BRANCH WHEN ALL ARE DONE
006130  000241                          CLC                             ;CLEAR CARRY FOR SHIFT
006132  006103                          ROL     R3                      ;SHIFT NEXT BIT TO CARRY
006134  103006                          BCC     13$                     ;BRANCH IF BIT NOT SET
006136  011100                          MOV     (R1),R0                 ;POINTER TO BIT DEFINITION
006140  112022                  11$:    MOVB    (R0)+,(R2)+             ;MOVE ASCIZ TO BUFFER
006142  001376                          BNE     11$                     ;MOVE ALL BITS
006144  112762  000054  177777          MOVB    #',,-1(R2)              ;INSERT A COMMA TO TERMINATE
006152  005721                  13$:    TST     (R1)+                   ;POINT TO NEXT DESCRIPTION
006154  000763                          BR      10$                     ;GET THE REMAINING BITS
006156  105042                  15$:    CLRB    -(R2)                   ;TERMINATE THE LINE
006160                                  PRINTX  #TSSDEF,#TMPBFR         ;PRINT THE BIT DEFINITIONS
006160  012746  002632                  MOV     #TMPBFR,-(SP)
006164  012746  006606                  MOV     #TSSDEF,-(SP)
```

PRITSSR - PRINT TSSR CONTENTS

```
                006170  012746  000002                  MOV     #2,-(SP)
                006174  010600                          MOV     SP,R0
                006176  104415                          TRAP    C$PNTX
                006200  062706  000006                  ADD     #6,SP
     1719
     1720       006204  010403              20$:        MOV     R4,R3           ;GET THE TSSR CONTENTS
     1721       006206  042703  177761                  BIC     #^CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
     1722       006212  016303  006676                  MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
     1723       006216                                  PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
                006216  010346                          MOV     R3,-(SP)
                006220  012746  006476                  MOV     #TCOASC,-(SP)
                006224  012746  000002                  MOV     #2,-(SP)
                006230  010600                          MOV     SP,R0
                006232  104415                          TRAP    C$PNTX
                006234  062706  000006                  ADD     #6,SP
     1724       006240  010403                          MOV     R4,R3           ;TSSR CONTENTS AGAIN
     1725       006242  042703  177717                  BIC     #^CFATERR,R3    ;CLEAR ALL BUT FATAL TERMINATION
     1726       006246  001416                          BEQ     25$             ;DON'T PRINT IF ZERO
     1727       006250  C06203                          ASR     R3
     1728       006252  006203                          ASR     R3
     1729       006254  006203                          ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
     1730       006256  016303  007236                  MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
     1731       006262                                  PRINTX  #TFCASC,R3      ;PRINT THE FATAL TERMINATION CODE
                006262  010346                          MOV     R3,-(SP)
                006264  012746  006537                  MOV     #TFCASC,-(SP)
                006270  012746  000002                  MOV     #2,-(SP)
                006274  010600                          MOV     SP,R0
                006276  104415                          TRAP    C$PNTX
                006300  062706  000006                  ADD     #6,SP
     1732       006304  042704  176377        25$:      BIC     #^CHIADDR,R4    ;CLEAR ALL BUT EXTENDED ADDRESS
     1733       006310  001411                          BEQ     30$             ;DON'T PRINT IF ZERO
     1734       006312                                  PRINTX  #TEXASC,R4      ;PRINT THE EXTENDED ADDRESS BITS
                006312  010446                          MOV     R4,-(SP)
                006314  012746  006435                  MOV     #TEXASC,-(SP)
                006320  012746  000002                  MOV     #2,-(SP)
                006324  010600                          MOV     SP,R0
                006326  104415                          TRAP    C$PNTX
                006330  062706  000006                  ADD     #6,SP
     1735       006334  013703  002200        30$:      MOV     EPRTSW,R3               ;PRINT MEAASGE BUFFER ADDRESS
     1736       006340                                  PRINTX  R3                      ;PRINT PROPER MESSAGE
                006340  010346                          MOV     R3,-(SP)
                006342  012746  000001                  MOV     #1,-(SP)
                006346  010600                          MOV     SP,R0
                006350  104415                          TRAP    C$PNTX
                006352  062706  000004                  ADD     #4,SP
     1737       006356  000207                          RTS     PC              ;RETURN TO CALLER
```

PRITSSR - PRINT TSSR CONTENTS

```
   1744 006360                                EPRT2:
   1745 006360     045     116     045  EPRT1:  .ASCIZ  '%N%A *****REPLACE M7196*****'
   1746
   1756 006415     045     116     045  TSSRFOR:        .ASCIZ  '%N%A TSSR = %O6'
   1757 006435     045     116     045  TEXASC:         .ASCIZ  '%N%A Extended Address Bits = %O6'
   1758 006476     045     116     045  TCOASC:         .ASCIZ  '%N%A Termination Class Code = %T'
   1759 006537     045     116     045  TFCASC:         .ASCIZ  '%N%A Fatal Termination Class Code = %T'
   1760 006606     045     116     045  TSSDEF:         .ASCIZ  '%N%A TSSR Bits Set: %T'
   1761 006635     045     116     045  AMBTSSR:        .ASCIZ  '%N%A TSSR Contents Are Ambiguous'
   1762                                        .EVEN
   1763 006676  006716  006741  006767  TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
   1764 006716     116     157     162  1$:     .ASCIZ  'Normal Termination'
   1765 006741     124     145     162  2$:     .ASCIZ  'Termination Condition'
   1766 006767     124     141     160  3$:     .ASCIZ  'Tape Status Alert'
   1767 007011     106     165     156  4$:     .ASCIZ  'Function Reject'
   1768 007031     122     145     143  5$:     .ASCIZ  'Recoverable Error - Tape Position One Record Down'
   1769 007113     122     145     143  6$:     .ASCIZ  'Recoverable Error - Tape Was Not Moved'
   1770 007162     125     156     162  7$:     .ASCIZ  'Unrecoverable Error'
   1771 007206     106     141     164  8$:     .ASCIZ  'Fatal Controller Error'
   1772                                        .EVEN
   1773
   1774 007236  007246  007302  007313  TSFCOD: .WORD   1$,2$,3$,4$
   1775 007246     111     156     164  1$:     .ASCIZ  'Internal Diagnostic Failure'
   1776 007302     122     145     163  2$:     .ASCIZ  'Reserved'
   1777 007313     102     165     163  3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
   1778 007357     122     145     163  4$:     .ASCIZ  'Reserved'
   1779                                        .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.
                                ;-

007370                          PRIPKT::
007370                                  SAVREG                          ;SAVE THE REGISTERS
007374  010005                          MOV     R0,R5                   ;SAVE NO. OF WORDS IN PACKET
007376  C05737  003134                  TST     KTENABLE                ;ABOVE 28K UNDER TEST?
007402  001001                          BNE     10$                     ;BR IF YES
007404  005003                          CLR     R3                      ;SET HIGH ORDER ADDRESS TO 0
007406  010301                  10$:    MOV     R3,R1                   ;COPY HIGH ORDER ADDRESS
007410  010400                          MOV     R4,R0                   ;GET LOWER ADDRESS
007412  006100                          ROL     R0                      ;SHIFT BIT 15 INTO C BIT
007414  006101                          ROL     R1                      ;AND INTO HIGH ORDER.
007416                                  PRINTB  #PKTADD,R1,R4           ;PRINT PACKET ADDRESS
007416  010446                          MOV     R4,-(SP)
007420  010146                          MOV     R1,-(SP)
007422  012746  007554                  MOV     #PKTADD,-(SP)
007426  012746  000003                  MOV     #3,-(SP)
007432  010600                          MOV     SP,R0
007434  104414                          TRAP    C$PNTB
007436  062706  000010                  ADD     #10,SP
007442  010300                  15$:    MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
007444  001404                          BEQ     20$                     ;BR IF NOT ABOVE 28K.
007446  010401                          MOV     R4,R1                   ;GET LOW ORDER ADDRESS
007450  004737  017316                  JSR     PC,SETMAP               ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
007454  010004                          MOV     R0,R4                   ;GET RETURNED PAR6 ADDRESS BIAS
007456  005001                  20$:    CLR     R1                      ;SAVE WORD NUMBER
007460  012402                  25$:    MOV     (R4)+,R2                ;GET PACKET CONTENTS
007462                                  PRINTB  #PKTFRM,R1,R2           ;PRINT THE DATA
007462  010246                          MOV     R2,-(SP)
007464  010146                          MOV     R1,-(SP)
007466  012746  007516                  MOV     #PKTFRM,-(SP)
007472  012746  000003                  MOV     #3,-(SP)
007476  010600                          MOV     SP,R0
007500  104414                          TRAP    C$PNTB
007502  062706  000010                  ADD     #10,SP
007506  005201                          INC     R1                      ;NEXT WORD NUMBER
007510  020105                          CMP     R1,R5                   ;DONE ALL PACKET WORDS?
007512  002762                          BLT     25$                     ;LOOP TILL ALL DONE
007514  000207                          RTS     PC                      ;RETURN

007516    045    116    045  PKTFRM: .ASCIZ  '%N%A Packet Word #%D1%A = %O6'
007554    045    116    045  PKTADD: .ASCIZ  '%N%A Packet Address = %O1%O5'
                                        .EVEN
```

```
                                        .SBTTL  PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

                                ;+
                                ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      XOR OF EXPECTED/RECEIVED DATA
                                ;-
007612                          PRIBXOR::
007612                                  SAVREG                          ;SAVE THE REGISTERS
007616  010203                          MOV     R2,R3                   ;EXPECTED DATA
007620                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007630  012700  177400                  MOV     #^C<377>,R0             ;BYTE MASK
007634  040001                          BIC     R0,R1                   ;SAVE LOW BYTE RECV
007636  040002                          BIC     R0,R2                   ;SAVE LOW BYTE EXPD
007640  040003                          BIC     R0,R3                   ;SAVE LOW BYTE XOR
007642                                  PRINTB  #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
007642  010346                          MOV     R3,-(SP)
007644  010146                          MOV     R1,-(SP)
007646  010246                          MOV     R2,-(SP)
007650  012746  007674                  MOV     #XORBFOR,-(SP)
007654  012746  000004                  MOV     #4,-(SP)
007660  010600                          MOV     SP,R0
007662  104414                          TRAP    C$PNTB
007664  062706  000012                  ADD     #12,SP
007670  010300                          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007672  000207                          RTS     PC                      ;RETURN TO CALLER

007674      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
                                ;-
007742                          PRIXOR::
007742                                  SAVREG                          ;SAVE THE REGISTERS
007746  010203                          MOV     R2,R3                   ;EXPECTED DATA
007750                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007760                                  PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
```

PRIXOR - PRINT EXPD, RECV AND XOR

```
        007760  010346                          MOV     R3,-(SP)
        007762  010146                          MOV     R1,-(SP)
        007764  010246                          MOV     R2,-(SP)
        007766  012746  010012                  MOV     #XORFOR,-(SP)
        007772  012746  000004                  MOV     #4,-(SP)
        007776  010600                          MOV     SP,R0
        010000  104414                          TRAP    C$PNTB
        010002  062706  000012                  ADD     #12,SP
  1873  010006  010300                          MOV     R3,R0           ;R0 HAS XOR ON RETURN
  1874  010010  000207                          RTS     PC              ;RETURN TO CALLER
  1875
  1876  010012     045     116     045  XORFOR: .ASCIZ  '<N><A EXPD: <06><A RECV: <06><A XOR: <06'
  1877                                          .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
                                ;
                                ;-

010060                          PRIEQU:
010060                                  SAVREG                          ;SAVE THE REGISTERS
010064  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
                                ;
                                ;-
010066                          PRIRAM:
010066                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
010072                                  PRINTB  @RAMFOR,R4      ;PRINT RAM ADDRESS IN ERROR
010072  010446                          MOV     R4,-(SP)
010074  012746  010116                  MOV     @RAMFOR,-(SP)
010100  012746  000002                  MOV     @2,-(SP)
010104  010600                          MOV     SP,R0
010106  104414                          TRAP    C$PNTB
010110  062706  000006                  ADD     @6,SP
010114  000207                          RTS     PC                      ;RETURN

010116     045     116     045  RAMFOR: .ASCIZ  '%N%A CONTROLLER RAM ADDRESS = %06'
                                        .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
                                ;
                                ;-
010160                          PRIADD:
010160                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010164  013700  002236                  MOV     ERRHI,R0                ;GET HIGH ADDRESSS
010170  013701  002240                  MOV     ERRLO,R1                ;GET LOW ADDRESS
010174  010102                          MOV     R1,R2                   ;COPY LOW ADDRESS
010176  006101                          ROL     R1                      ;SHIFT BIT 15 TO C BIT
010200  006100                          ROL     R0                      ;SHIFT INTO HIGH ORDER
010202                                  PRINTB  #PRIA0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
010202  010246                          MOV     R2,-(SP)
010204  010046                          MOV     R0,-(SP)
010206  012746  010230                  MOV     #PRIA0,-(SP)
010212  012746  000003                  MOV     #3,-(SP)
010216  010600                          MOV     SP,R0
010220  104414                          TRAP    C$PNTB
010222  062706  000010                  ADD     #10,SP
010226  000207                          RTS     PC                      ;RETURN

010230  045  116  045  PRIA0:  .ASCIZ  '%N%A MEMORY ERROR ADDRESS = %01%05'
                             .EVEN

                                        .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
                                ;*
                                ;
                                ;PRINT MEMORY ADDRESS
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ; IMPLICIT INPUTS
                                ;
                                ;       ERRHI   - HIGH ORDER ADDRESS
                                ;       ERRLO   - LOW ORDER ADDRESS
                                ;
                                ;-
010274                          PRITADD:
010274                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010300  013702  002236                  MOV     ERRHI,R2                ;GET HIGH ADDRESSS
010304  013701  002240                  MOV     ERRLO,R1                ;GET LOW ADDRESS
                                        ;MOV    R1,R2                   ;COPY LOW ADDRESS
                                        ;ROL    R1                      ;SHIFT BIT 15 TO C BIT
                                        ;ROL    R0                      ;SHIFT INTO HIGH ORDER
010310                                  PRINTB  #PRIT0,R1               ;PRINT MEMORY ADDRESS LOW IN ERROR
010310  010146                          MOV     R1,-(SP)
010312  012746  010356                  MOV     #PRIT0,-(SP)
010316  012746  000002                  MOV     #2,-(SP)
010322  010600                          MOV     SP,R0
010324  104414                          TRAP    C$PNTB
```

PRITADD - PRINT MEMORY TEST ADDRESS

```
        010326  062706  000006                  ADD     #6,SP
   1961 010332                                  PRINTB  @PRIT1,R2       ;PRINT MEMORY ADDRESS HIGH IN ERROR
        010332  010246                          MOV     R2,-(SP)
        010334  012746  010421                  MOV     #PRIT1,-(SP)
        010340  012746  000002                  MOV     #2,-(SP)
        010344  010600                          MOV     SP,R0
        010346  104414                          TRAP    C$PNTB
        010350  062706  000006                  ADD     #6,SP
   1962 010354  000207                          RTS     PC              ;RETURN
   1963
   1964 010356     045     116     045  PRIT0:  .ASCIZ  '<N><A MEMORY TEST ADDRESS LOW = <06>'
   1965 010421     045     116     045  PRIT1:  .ASCIZ  '<N><A MEMORY TEST ADDRESS HIGH = <06>'
   1966                                         .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:
                                ;
                                ;
                                ;-

010466                          SPACE::
010466                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
010472  012737  000764  010660          MOV     #500.,SDELAY            ;SET UP DELAY
010500  012737  140010  010650          MOV     #140010,80$             ;SET UP COMMAND, SPACE FORWARD
010506  005703                          TST     R3                      ;CHECK FOR DIRECTION
010510  100403                          BMI     5$                      ;BR, IF REVERSE INDICATED
010512  010337  010652                  MOV     R3,90$                  ;LOAD UP NUMBER OF RECORDS TO SPACE
010516  000407                          BR      10$                     ;GO DO COMMAND
010520  042703  100000          5$:     BIC     #BIT15,R3               ;CLEAR DIRECTION BIT
010524  010337  010652                  MOV     R3,90$                  ;LOAD UP NUMBER OF RECORDS TO SPACE
010530  052737  000400  010650          BIS     #BIT8,80$               ;SET REVERSE BIT IN COMMAND PACKET
010536  012704  010650          10$:    MOV     #80$,R4                 ;SET UP R4 WITH PACKET ADDRESS
010542  010465  000000                  MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010546  004737  016250          15$:    JSR     PC,WAITF                ;WAIT FOR SSR
010552  103420                          BCS     20$                     ;BR, IF SSR IS SET AND OK
010554                                  DELAY   250             ;DELAY ABOUT .25 SECONDS
010554  012727  000250                  MOV     #250.,(PC)+
010560  000000                          .WORD   0
010562  013727  002116                  MOV     L$DLY,(PC)+
010566  000000                          .WORD   0
010570  005367  177772                  DEC     -6(PC)
010574  001375                          BNE     .-4
```

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```
        010576  005367  177756                  DEC     -22(PC)
        010602  001367                          BNE     .-20
   2019 010604  005337  010660                  DEC     SDELAY                  ;BUMP DELAY COUNTER DOWN
   2020 010610  001356                          BNE     15$                     ;BR. IF MORE DELAY
   2021 010612  000411                          BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
   2022 010614  016501  000002          20$:    MOV     TSSR(R5),R1             ;READ TSSR
   2023 010620  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
   2024 010624  020201                  25$:    CMP     R2,R1                   ;ARE THEY OK
   2025 010626  001401                          BEQ     40$                     ;BR. IF EQUAL = OK
   2026 010630  000402                          BR      60$                     ;TROUBLE EXIT
   2027 010632  000261                  40$:    SEC                             ;SET CARRY NO TROUBLE
   2028 010634  000401                          BR      70$                     ;EXIT
   2029 010636  000241                  60$:    CLC                             ;CARRY CLEAR = ERROR
   2030 010640                          70$:
   2031 010640  010400                          MOV     R4,R0                   ;PASS PACKET ADDRESS
   2032 010642  000207                          RTS     PC                      ;RETURN
   2033                                 ;
   2034                                 ;
   2035                                 ;
   2036                                 ;PACKET FOR SPACE COMMAND
   2037                                 ;
   2039         010650                          .=<.+10>&177770
   2041                                 ;
   2042                                 ;COMMAND WORD
   2043 010650  000000                  80$:    .WORD
   2044                                 ;NUMBER OF RECORDS TO BE SPACED OVER WORD
   2045 010652  000000                  90$:    .WORD
   2046 010654  000000                          .WORD
   2047 010656  000000                          .WORD
   2048 010660  000000                  SDELAY: .WORD   0                       ;DELAY COUNTER
   2049                                         .EVEN
   2050                                         .SBTTL  WRTCHR  - WRITE CHARACTERISTICS COMMAND
```

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:
                                        ;               EXTFEA = EXTENDED FEATURES PRESENT
                                        ;               BENBSW = BUFFER ENABLE SWITCH ON OR OFF
                                        ;
                                        ;SIDE EFFECTS:
                                        ;-
010662                                  WRTCHR::
010662                                          SAVREG                          ;SAVE THE GENERAL REGISTERS
010666  005037  002230                          CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010672  005037  002226                          CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SW SWITCH
010676  010465  000000          10$:    MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010702  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
010706  103401                          BCS     20$                     ;BR, IF SSR IS SET AND OK
010710  000435                          BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
010712  016501  000002          20$:    MOV     TSSR(R5),R1             ;READ TSSR
010716  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
010722  032701  000100                  BIT     #OFL,R1                 ;WAS OFF LINE SET IN TSSR
010726  001402                          BEQ     25$                     ;BR, IF NO OFL SET
010730  052702  000100                  BIS     #OFL,R2                 ;MAKE THEM LOOK ALIKE
010734  020201                  25$:    CMP     R2,R1                   ;ARE THEY OK
010736  001401                          BEQ     40$                     ;BR, IF EQUAL = OK
010740  000421                          BR      60$                     ;TROUBLE EXIT
010742  062704  000010          40$:    ADD     #8.,R4                  ;POINT TO WRT CHARA DATA PACKET
010746  011403                          MOV     (R4),R3                 ;GET ADDRESS OF MESSAGE BUFFER
010750  032763  000200  000012          BIT     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURES BIT SET?
010756  001402                          BEQ     45$                     ;BR IF NO
010760  005237  002226                  INC     EXTFEA                  ;SET EXTENDED FEATURES SW SWITCH
010764                          45$:
010764  032763  000100  000012          BIT     #X2.BUFE,XST2(R3)       ;BUFFER ENABLE SWITCH SET
010772  001402                          BEQ     50$                     ;BR, IF SWITCH NOT SET
010774  005237  002230                  INC     BENBSW                  ;SET SOFTWARE SWITCH FOR ENABLED
011000                          50$:
011000  000261                          SEC                             ;SET CARRY NO TROUBLE
011002  000401                          BR      70$                     ;EXIT
011004  000241                  60$:    CLC                             ;CARRY CLEAR = ERROR
011006  016500  000002          70$:    MOV     TSSR(R5),R0             ;RETURN TSSR CONTENTS
011012  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
                                ;
                                ;
                                ;-
011014                          REWIND::
011014                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011020  012704  011110                  MOV     #RWPACK,R4              ;GET PACKET ADDRESS
011024  010465  000000                  MOV     R4,TSDB(R5)             ;SEND PACKET ADDRESS TO EXECUTE
011030  012703  000550                  MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
011034  004737  016250          10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
011040  103417                          BCS     20$                     ;LEAVE WHEN SSR IS SET
011042                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
011042  012727  000372                  MOV     #250.,(PC)+
011046  000000                          .WORD   0
011050  013727  002116                  MOV     L$DLY,(PC)+
011054  000000                          .WORD   0
011056  005367  177772                  DEC     -6(PC)
011062  001375                          BNE     .-4
011064  005367  177756                  DEC     -22(PC)
011070  001367                          BNE     .-20
011072  005303                          DEC     R3                      ;BUMP COUNTER DOWN
011074  001357                          BNE     10$                     ;KEEP GOING
011076  000241                          CLC                             ;CLEAR CARRY TO SET ERROR
011100  010400          20$:    MOV     R4,R0                           ;PASS THE PACKET ADDRESS
011102  000207                  RTS     PC                              ;RETURN
                                
011110                                  .=<.+10>&177770
011110                          RWPACK:
011110  102010                          .WORD   102010                  ;POSTION COMMAND (REWIND)
011112  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:
                                ;
                                ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
                                ;
                                ;-

011114                          CKRAM::
011114                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
011120  012701  002242                  MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011124  012702  000201                  MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
011130  005003                          CLR     R3                      ;CLEAR THE ERROR FLAG
011132  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
011136  112765  000000  000000          MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011144  004737  016336          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011150  010265  000000                  MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011154  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011160  116511  000000                  MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011164  122124                          CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011166  001401                          BEQ     20$                     ;BRANCH IF OK
011170  005203                          INC     R3                      ;SET ERROR FLAG
011172  005202                  20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011174  020227  000210                  CMP     R2,#RMPKTEND            ;REACHED END YET ?
011200  003761                          BLE     10$                     ;BRANCH TILL ALL READ
011202  005703                          TST     R3                      ;WAS AN ERROR FOUND ?
011204  001402                          BEQ     30$                     ;BRANCH IF NOT
011206  000241                          CLC                             ;CLEAR CARRY TO SHOW ERROR
011210  000401                          BR      50$                     ;AND EXIT
011212  000261                  30$:    SEC                             ;SHOW GOOD COMPARE
011214  012737  000010  002302  50$:    MOV     #8.,RAMSIZ              ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
011222  00^207                          RTS     PC                      ;RETURN
```

```
                                        .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:
                                ;
                                ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
                                ;-
011224                          CKRAM2::
                                        SAVREG                          ;SAVE THE GENERAL REGISTERS
011230  012701  002242                  MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011234  012702  000167                  MOV     #RMCHBEG,R2             ;BYTE ADDRESS OF FIRST RAM DATA
011240  005003                          CLR     R3                      ;CLEAR THE ERROR FLAG
011242  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
011246  112765  000000  000000          MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011254  004737  016336          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011260  010265  000000                  MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011264  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011270  116511  000000                  MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011274  122124                          CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011276  001401                          BEQ     20$                     ;BRANCH IF OK
011300  005203                          INC     R3                      ;SET ERROR FLAG
011302  005202                  20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011304  012737  000010  002302          MOV     #8.,RAMSIZ              ;ASSUME EXTFEA NOT SET
011312  005737  002226                  TST     EXTFEA                  ;IS THE SOFTWARE EXTENDED FEATURES SET
011316  001407                          BEQ     25$                     ;BR. IF NOT SET
011320  012737  000012  002302          MOV     #10.,RAMSIZ             ;SET RAMSIZ FOR EXTEND FEATURES
011326  020227  000200                  CMP     R2,#RMCHEND             ;AT END OF EXTENDED BUFFER
011332  003750                          BLE     10$                     ;BR. IF NOT AT END YET
011334  000403                          BR      27$                     ;AT END BRANCH
011336  020227  000176          25$:    CMP     R2,#RMCHEND-2           ;REACHED END YET ?
011342  003744                          BLE     10$                     ;BRANCH TILL ALL READ
011344  005703                  27$:    TST     R3                      ;WAS AN ERROR FOUND ?
011346  001402                          BEQ     30$                     ;BRANCH IF NOT
011350  000241                          CLC                             ;CLEAR CARRY TO SHOW ERROR
011352  000401                          BR      50$                     ;AND EXIT
011354  000261                  30$:    SEC                             ;SHOW GOOD COMPARE
011356  000207                  50$:    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
                                ;
                                ;-
011360                          CKMSG::
011360                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
011364  010037  002304                  MOV     R0,RCVHIADD     ;SAVE RECV HIGH ADDRESS
011370  010137  002306                  MOV     R1,RCVLOAD      ;SAVE RECV LOW ADDRESS
011374  005737  003134                  TST     KTENABLE        ;TESTING ABOVE 28K?
011400  001403                          BEQ     10$             ;BR IF NO
011402  004737  017316                  JSR     PC,SETMAP       ;RETURN ADDRESS BIASED TO PAR6 IN R0
011406  010001                          MOV     R0,R1           ;GET RETURNED ADDRESS BIASED TO PAR6
011410  005004                  10$:    CLR     R4              ;WORD IN BUFFER
011412  005003                          CLR     R3              ;CLEAR ERROR SEEN FLAG
011414  010205                          MOV     R2,R5           ;GET EXPD BUFFER ADDRESS
011416  011264  002322          15$:    MOV     (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
011422  011164  002466                  MOV     (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
011426  022221                          CMP     (R2)+,(R1)+     ;EXPD EQUAL RECV?
011430  001401                          BEQ     25$             ;BR IF YES
011432  005203                          INC     R3              ;SET ERROR SEEN FLAG
011434  062704  000002          25$:    ADD     #2,R4           ;POINT TO NEXT WORD ADDRESS
011440  020427  000014                  CMP     R4,#14          ;DONE FIRST 7 WORDS?
011444  003764                          BLE     15$             ;BR IF NO
011446  032765  000200  000012          BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
011454  001403                          BEQ     50$             ;BR IF NO
011456  020427  000016                  CMP     R4,#16          ;DONE EXTENDED FEATURES WORD?
011462  003755                          BLE     15$             ;BR IF NO
011464  005703                  50$:    TST     R3              ;ANY ERRORS SEEN?
011466  001402                          BEQ     55$             ;BR IF NO
011470  000241                          CLC                     ;SET FAILURE
011472  000401                          BR      60$             ;
011474  000261                  55$:    SEC                     ;SET SUCCESS
011476  000207                  60$:    RTS     PC              ;RETURN
```

```
                                        .SBTTL  CKMSG2  - COMPARE EXPD RECV MESSAGE BUFFERS
                                ;+
                                ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                                ;ERROR PRINT ROUTINES.
                                ;
                                ;INPUT:
                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
                                ;       R3      NUMBER OF BYTES TO COMPARE
                                ;
                                ;OUTPUT:
                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
                                ;               CLR - MESSAGE BUFFERS DON'T MATCH
                                ;
                                ;IMPLICIT OUTPUT:
                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
                                ;       RECMSG          BUFFER IS SET TO RECV DATA
                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                                ;-
011500                          CKMSG2::
011500                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011504  020327  000144                   CMP     R3,#RECMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
011510  003412                           BLE     5$                      ;@@D BR IF NO
011512  012703  000144                   MOV     #RECMSG-EXPMSG,R3;@@D
011516                                  PRINTF  #DEBUGMSG               ;@@D
011516  012746  011632                   MOV     #DEBUGMSG,-(SP)
011522  012746  000001                   MOV     #1,-(SP)
011526  010600                           MOV     SP,R0
011530  104417                           TRAP    C$PNTF
011532  062706  000004                   ADD     #4,SP
011536  010037  002304           5$:     MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
011542  010137  002306                   MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
011546  005737  003134                   TST     KTENABLE                ;TESTING ABOVE 28K?
011552  001403                           BEQ     10$                     ;BR IF NO
011554  004737  017316                   JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
011560  010001                           MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
011562  005004           10$:    CLR     R4                      ;WORD IN BUFFER
011564  005005                   CLR     R5                      ;CLEAR ERROR SEEN FLAG
011566  111264  002322   15$:    MOVB    (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
011572  111164  002466           MOVB    (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
011576  122221                   CMPB    (R2)+,(R1)+             ;EXPD EQUAL RECV?
011600  001401                   BEQ     25$                     ;BR IF YES
011602  005205                   INC     R5                      ;SET ERROR SEEN FLAG
011604  062704  000001   25$:    ADD     #1,R4                   ;POINT TO NEXT BYTE
011610  020403                   CMP     R4,R3                   ;DONE ALL BYTES?
011612  002001                   BGE     50$                     ;BR IF YES
011614  000764                   BR      15$                     ;DO NEXT BYTE
011616  005705           50$:    TST     R5                      ;ANY ERRORS SEEN?
011620  001402                   BEQ     55$                     ;BR IF NO
011622  000241                   CLC                             ;SET FAILURE
011624  000401                   BR      60$                     ;
011626  000261           55$:    SEC                             ;SET SUCCESS
011630  000207           60$:    RTS     PC                      ;RETURN
```

```
011632     120     122     117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';88D
011722     045     116     045  FERCM:   .ASCII  /%N%A  ***/
011733     040     040     124  ERCM:    .ASCIZ  /  TSSR ERROR CODE REC'D = /
011766     056     056     056  SIMSG:   .ASCIZ  /.... AFTER DOING SOFT INIT/
012021     124     105     123  TINERR:  .ASCIZ  /TEST: .../
                                         .EVEN
                                ;+
                                ;
                                ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
                                ;
                                ;INPUT:
                                ;
                                ;        R1      CONTENTS OF TSSR AT ERROR
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;        EXECUTES DROP UNIT TO CEASE TESTING
                                ;
                                ;-

012034                                   BGNMSG  SFIMSG
012034                          SFIMSG::
012034  004737  006024                   JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
012040  004737  017202                   JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
012044                                   ENDMSG
012044                          L10003:
012044  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
                                ;
                                ;-

012046                                   BGNMSG  PKTSSR
012046                          PKTSSR::
012046  004737  006024                   JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012052  012700  000004                   MOV     #4,R0           ;NO. OF WORDS IN PACKET
012056  004737  007370                   JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
012062                                   ENDMSG
012062                          L10004:
012062  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
012064                                  PKTGETS::
012064  004737  006024                          JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012070  012700  000002                          MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
012074  004737  007370                          JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
012100                                          ENDMSG
012100                                  L10005:
012100  104423                                  TRAP    C$MSG
                                        ;+
                                        ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
                                        ;
                                        ;INPUTS:
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;-
                                                BGNMSG  SFFMSG
012102                                  SFFMSG::
012102  004737  006024                          JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
012106                                          ENDMSG
012106                                  L10006:
012106  104423                                  TRAP    C$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
012110                                  PKTMES::
012110  004737  006024                          JSR     PC,PRITSSR              ;PRINT CONTENTS OF TSSR
012114  010200                                  MOV     R2,R0                   ;LOW ORDER ADDRESS
012116  010301                                  MOV     R3,R1                   ;HIGH ORDER ADDRESS
012120  004737  014242                          JSR     PC,PRMESS               ;PRINT THE MESSAGE BUFFER
012124                                          ENDMSG
012124                                  L10007:
012124  104423                                  TRAP    C$MSG
```

```
                                        .SBTTL  ADDSSR  - PRINT TEST ADDRESS AN' TSSR
                                ;+
                                ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                ;TSSR AND A MEMORY TEST ADDRESS
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       ERRHI   HIGH ORDER MEMORY TEST ADDRESS
                                ;       ERRLO   LOW ORDER MEMORY TEST ADDRESS
                                ;-
                                        BGNMSG  ADDSSR
012126                          ADDSSR::
012126  004737  010274                  JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
012132  016501  000002                  MOV     TSSR(R5),R1     ;GET CURRENT TSSR
012136  004737  006024                  JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012142                                  ENDMSG
012142                          L10010:
012142  104423                          TRAP    C$MSG

                                        .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
                                        BGNMSG  MSGEXP
012144                          MSGEXP::
012144  012700  000007                  MOV     #7,R0           ;ASSUME NO EXT FEATURES
012150  005737  002226                  TST     EXTFEA          ;EXT FEATURES SET?
012154  001402                          BEQ     5$              ;BR IF NO
012156  012700  000010                  MOV     #8.,R0          ;EXT FEATURE BUFFER IS 8 WORDS
012162  004737  014552          5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012166                                  ENDMSG
012166                          L10011:
012166  104423                          TRAP    C$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
012170                          FIFEXP::
012170                                  PRINTX  #FIF1MSG,R1     ;PRINT BYTES TRANSFERRED
012170  010146                          MOV     R1,-(SP)
012172  012746  012242                  MOV     #FIF1MSG,-(SP)
012176  012746  000002                  MOV     #2,-(SP)
012202  C10600                          MOV     SP,R0
012204  104415                          TRAP    C$PNTX
012206  062706  000006                  ADD     #6,SP
012212                                  PRINTX  #FIF2MSG        ;PRINT HEADER MSG
012212  012746  012311                  MOV     #FIF2MSG,-(SP)
012216  012746  000001                  MOV     #1,-(SP)
012222  010600                          MOV     SP,R0
012224  104415                          TRAP    C$PNTX
012226  062706  000004                  ADD     #4,SP
012232  010100                          MOV     R1,R0           ;GET BYTE COUNT
012234  004737  015122                  JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
012240                                  ENDMSG
012240                          L10012:
012240  104423                          TRAP    C$MSG
012242     045     116     045  FIF1MSG:        .ASCIZ  '<15><12>A NUMBER OF BYTES TRANSFERRED = <D2>'
012311     045     116     045  FIF2MSG:        .ASCIZ  '<15><12>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
                                ;-
012350                                  BGNMSG  MSGSTAT
012350                          MSGSTAT::
012350  012701  012412                    MOV     #STATCOD,R1     ;ASCII ADDRESS TABLE
012354  012100                  10$:      MOV     (R1)+,R0        ;DONE ALL MSG LINES?
012356  001410                            BEQ     20$             ;BR IF YES
012360                                    PRINTX  R0              ;PRINT STATUS BIT NAMES
012360  010046                            MOV     R0,-(SP)
012362  012746  000001                    MOV     #1,-(SP)
012366  010600                            MOV     SP,R0
012370  104415                            TRAP    C$PNTX
012372  062706  000004                    ADD     #4,SP
012376  000766                            BR      10$             ;DO ANOTHER MSG LINE
012400  012700  000012          20$:      MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
012404  004737  014552                    JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012410                                    ENDMSG
012410                          L10013:
012410  104423                            TRAP    C$MSG

012412  012430  012472  012563  STATCOD:        .WORD   1$,2$,3$,4$,5$,6$,0
012430    045     116     045   1$:.ASCIZ  '<15><12>A Tape Bus Signals in Word #8:'
012472    045     116     045   2$:.ASCIZ  '<15><12>A        PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>'
012563    045     116     045   3$:.ASCIZ  '<15><12>A        IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>'
012654    045     116     045   4$:.ASCIZ  '<15><12>A        IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>'
012745    045     116     045   5$:.ASCIZ  '<15><12>A Tape Bus Signals in Word #9:'
013007    045     116     045   6$:.ASCIZ  '<15><12>A        DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>'
                                        .EVEN
```

MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

```
                                        .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
                                ;-
013064                                  BGNMSG  MSGLOOP
013064                          MSGLOOP::
013064  012701  013126                  MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
013070  012100                  10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
013072  001410                          BEQ     20$             ;BR IF YES
013074                                  PRINTX  R0              ;PRINT STATUS BIT NAMES
013074  C10046                          MOV     R0,-(SP)
013076  012746  000001                  MOV     #1,-(SP)
013102  010600                          MOV     SP,R0
013104  104415                          TRAP    C$PNTX
013106  062706  000004                  ADD     #4,SP
013112  000766                          BR      10$             ;DO ANOTHER MSG LINE
013114  012700  000012          20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
013120  004737  014552                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013124                                  ENDMSG
013124                          L10014:
013124  104423                          TRAP    C$MSG

013126  013146  013221  013320  LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
013146     045     116     045  1$:.ASCIZ  '<NL>A Tape Bus Loopback Signals in Word #0:'
013221     045     116     045  2$:.ASCIZ  '<NL>A      PARERR<15>          IRESV2<14>          IRESV1<13>'
013320     045     116     045  3$:.ASCIZ  '<NL>A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
013417     045     116     045  4$:.ASCIZ  '<NL>A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
013516     045     116     045  5$:.ASCIZ  '<NL>A ITADO=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
013615     045     116     045  6$:.ASCIZ  '<NL>A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
013714     045     116     045  7$:.ASCIZ  '<NL>A IGO  =>IFPT<00>'
                                        .EVEN
```

```
                                        .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
                                ;-
013742                                  BGNMSG  MSGSUB
013742                          MSGSUB::
013742  012700  000012                  MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
013746  004737  014552                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013752                                  ENDMSG
013752                          L10015:
013752  104423                          TRAP    C$MSG
                                        .SBTTL  MEMADD - PRINT MEMORY ADDRESS DATA ERROR
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       ERRHI   - MEMORY ERROR HIGH ORDER ADDRESS
                                ;       ERRLO   - MEMORY ERROR LOW ORDER ADDRESS
                                ;       EXP     - EXPECTED DATA
                                ;       RECV    - RECEIVED DATA
                                ;-
013754                                  BGNMSG  MEMADD
013754                          MEMADD::
013754  004737  010160                  JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
013760  013701  002232                  MOV     EXPD,R1         ;GET EXPD DATA
013764  013702  002234                  MOV     RECV,R2         ;GET RECEIVED DATA
013770  004737  007742                  JSR     PC,PRIXOR       ;PRINT EXPD/RECV
013774                                  ENDMSG
013774                          L10016:
013774  104423                          TRAP    C$MSG
```

```
                                        .SBTTL  PRAMPKT - PRINT RAM AND PACKET DATA
                                ;+
                                ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                ;WHEN THE RAM DATA DOES NOT MATCH.
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      POINTER TO COMMAND PACKET
                                ;IMPLICIT INPUTS:
                                ;       RAMDATA         DATA AS READ FROM THE RAM
                                ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;       RAMSIZ  SET TO 0
                                ;-
013776                          PRAMPKT:
013776                                  SAVREG                                  ;SAVE R1-R5 UNTIL NEXT RETURN
014002  C12701  002242                  MOV     @RAMDATA,R1                     ;DATA FROM THE RAM
014006  005002                          CLR     R2                              ;INIT BYTE NUMBER
014010  122124                  5$:     CMPB    (R1)+,(R4)+                     ;COMPARE EXPECTED, RECEIVED
014012  001005                          BNE     7$                              ;BR IF NO MATCH
014014                                  FORCERROR       7$,NOTSSR
014024  000436                          BR      10$                             ;BRD
014026  116105  177777          7$:     MOVB    -1(R1),R5                       ;GET RECV RAM DATA
014032  116403  177777                  MOVB    -1(R4),R3                       ;GET EXPD PACKET DATA
014036                                  XOR     R5,R3                           ;XOR EXPD/RECV
014046  042703  177400                  BIC     @177400,R3                      ;LOW BYTE ONLY
014052  116137  177777  002234          MOVB    -1(R1),RECV                     ;GET RECEIVED RAM DATA
014060  116437  177777  002232          MOVB    -1(R4),EXPD                     ;GET EXPECTED RAM DATA
014066                                  PRINTB  @RAMASC,R2,RECV,EXPD,R3
014066  010346                          MOV     R3,-(SP)
014070  013746  002232                  MOV     EXPD,-(SP)
014074  013746  002234                  MOV     RECV,-(SP)
014100  010246                          MOV     R2,-(SP)
014102  012746  014156                  MOV     @RAMASC,-(SP)
014106  012746  000005                  MOV     @5,-(SP)
014112  010600                          MOV     SP,R0
014114  104414                          TRAP    C$PNTB
014116  062706  000014                  ADD     @14,SP
014122  005202                  10$:    INC     R2                              ;UPDATE BYTE COUNT
014124  005737  002302                  TST     RAMSIZ                          ;DEFAULT TO 8.?
014130  001404                          BEQ     15$                             ;BR IF YES
014132  020237  002302                  CMP     R2,RAMSIZ                       ;DONE ALL BYTES?
014136  003724                          BLE     5$                              ;BR IF NO
014140  000403                          BR      25$                             ;
014142  020227  000010          15$:    CMP     R2,@8.                          ;DONE DEFAULT NUMBER OF BYTES?
014146  002720                  20$:    BLT     5$                              ;BR IF NO
014150  005037  002302          25$:    CLR     RAMSIZ                          ;SET DEFAULT RAMSIZ
014154  000207                          RTS     PC                              ;RETURN

014156      045     116     045 RAMASC: .ASCIZ  '%N%A  BYTE: %D2%A RAM: %03%A Packet: %03%A XOR:%03'
                                        .EVEN
```

```
                                        .SBTTL  PRMESS  - PRINT CONTENTS OF MESSAGE BUFFER
                                ;+
                                ;THIS ROUTINE PRINTS THE CONTENTS OF
                                ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE TSV-05.
                                ;
                                ;INPUT:
                                ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
                                ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
                                        NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
                                ;-
2675 014242                     PRMESS: SAVREG                  ;SAVE THE REGISTERS
2676 014246  010005                     MOV     R0,R5           ;SAVE LOW ORDER ADDRESS
2677 014250  005737  003134             TST     KTENABLE        ;ADDRESS ABOVE 28K?
2678 014254  001001                     BNE     10$             ;BR IF YES
2679 014256  005001                     CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
2680 014260  010103             10$:    MOV     R1,R3           ;SAVE HIGH ORDER ADDRESS
2681 014262  006100                     ROL     R0              ;SHIFT BIT15 TO C BIT
2682 014264  006101                     ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
2683 014266                             PRINTX  #PROASC,R1,R5   ;PRINT MESSAGE BUFFER ADDRESS
     014266  010546                     MOV     R5,-(SP)
     014270  010146                     MOV     R1,-(SP)
     014272  012746  014420             MOV     #PROASC,-(SP)
     014276  012746  000003             MOV     #3,-(SP)
     014302  010600                     MOV     SP,R0
     014304  104415                     TRAP    C$PNTX
     014306  062706  000010             ADD     #10,SP
2684 014312                             PRINTX  #PR1ASC         ;PRINT HEADER FOR CONTENTS
     014312  012746  014465             MOV     #PR1ASC,-(SP)
     014316  012746  000001             MOV     #1,-(SP)
     014322  010600                     MOV     SP,R0
     014324  104415                     TRAP    C$PNTX
     014326  062706  000004             ADD     #4,SP
2685 014332  005004                     CLR     R4              ;NUMBER OF THE NEXT WORD
2686 014334  010501                     MOV     R5,R1           ;COPY LOW ORDER ADDRESS
2687 014336  010300                     MOV     R3,R0           ;COPY HIGH ORDER ADDRESS
2688 014340  001403                     BEQ     20$             ;BR IF NOT ABOVE 28K
2689 014342  004737  017316             JSR     PC,SETMAP       ;SETUP PAR ADDRESS IN R0
2690 014346  010005                     MOV     R0,R5           ;GET PAR FORMAT ADDRESS ABOVE 28K
2691 014350                     20$:    PRINTX  #PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
     014350  012546                     MOV     (R5)+,-(SP)
     014352  010446                     MOV     R4,-(SP)
     014354  012746  014523             MOV     #PRASC,-(SP)
     014360  012746  000003             MOV     #3,-(SP)
     014364  010600                     MOV     SP,R0
     014366  104415                     TRAP    C$PNTX
     014370  062706  000010             ADD     #10,SP
2692 014374  005204                     INC     R4              ;NUMBER OF THE NEXT
2693 014376  020427  000007             CMP     R4,#7           ;DONE ALL YET ?
2694 014402  003005                     BGT     50$             ;BRANCH IF ALL DONE
2695 014404  002761                     BLT     20$             ;PRINT FIRST 7 WORDS
2696 014406  032763  000200  000012     BIT     #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
2697 014414  001355                     BNE     20$             ;PRINT EXTENDED STATUS WORD
2698 014416  000207                     RTS     PC              ;RETURN
2699 014420     045     116     045     PROASC: .ASCIZ  '<N><A Message Buffer Address = <01><05>'
2700 014465     045     116     045     PR1ASC: .ASCIZ  '<N><A Message Buffer Contents:'
2701 014523     045     116     045     PRASC:  .ASCIZ  '<N><A    Word<D1><A: <0>'
```

```
                                        .EVEN
                                        .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                                ;+
                                ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
                                ;       R0      - NUMBER OF WORDS IN BUFFER
                                ;IMPLICIT INPUTS:
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
014552                          PRMSGEXP::
014552                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014556  010005                          MOV     R0,R5                   ;SAVE NUMBER OF WORDS
014560  013700  002306                  MOV     RCVLOADD,R0             ;GET RECV LOW ADDRESS
014564  010004                          MOV     R0,R4                   ;COPY LOW ADDRESS
014566  013701  002304                  MOV     RCVHIADD,R1             ;GET RECV HIGH ADDRESS
014572  006100                          ROL     R0                      ;SHIFT BIT15 TO C BIT
014574  006101                          ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
014576                                  PRINTX  #PRMSG0,R1,R4           ;PRINT MESSAGE BUFFER ADDRESS
014576  010446                          MOV     R4,-(SP)
014600  010146                          MOV     R1,-(SP)
014602  012746  01[illegible]32          MOV     #PRMSG0,-(SP)
014606  012746  000003                  MOV     #3,-(SP)
014612  010600                          MOV     SP,R0
014614  104415                          TRAP    C$PNTX
014616  062706  000010                  ADD     #10,SP
014622                                  PRINTX  #PRMSG1                 ;PRINT HEADER FOR CONTENTS
014622  012746  014777                  MOV     #PRMSG1,-(SP)
014626  012746  000001                  MOV     #1,-(SP)
014632  010600                          MOV     SP,R0
014634  104415                          TRAP    C$PNTX
014636  062706  000004                  ADD     #4,SP
014642  005004                          CLR     R4                      ;NUMBER OF THE CURRENT WORD
014644  012701  002322                  MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
014650  012702  002466                  MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
014654  011100                  20$:    MOV     (R1),R0                 ;GET EXPD
014656  011203                          MOV     (R2),R3                 ;GET RECV
014660                                  XOR     R0,R3                   ;XOR EXPD/RECV
014670                                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
014670  010346                          MOV     R3,-(SP)
014672  012246                          MOV     (R2)+,-(SP)
014674  012146                          MOV     (R1)+,-(SP)
014676  010446                          MOV     R4,-(SP)
014700  012746  015035                  MOV     #PRMSG2,-(SP)
014704  012746  000005                  MOV     #5,-(SP)
014710  010600                          MOV     SP,R0
014712  104415                          TRAP    C$PNTX
014714  062706  000014                  ADD     #14,SP
014720  005204                          INC     R4                      ;NUMBER OF THE NEXT
014722  020405                          CMP     R4,R5                   ;DONE ALL YET?
014724  002001                          BGE     50$                     ;BR IF YES
014726  000752                          BR      20$                     ;DO ANOTHER
014730  000207                  50$:    RTS     PC                      ;RETURN
014732     045     116     045  PRMSG0: .ASCIZ   '<N><A Message Buffer Address = <01><05>'
014777     045     116     045  PRMSG1: .ASCIZ   '<N><A Message Buffer Contents:'
015035     04      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
                                ;-
2754 015122                     PRBYTEXP::
2755 015122                             SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
2756 015126  010005                     MOV     R0,R5                   ;SAVE NUMBER OF BYTES
2757 015130  005037  002320             CLR     PRMNO                   ;INIT ERROR COUNT
2758 015134  C05004                     CLR     R4                      ;NUMBER OF THE CURRENT BYTE
2759 015136  012701  002322             MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
2760 015142  012702  002466             MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
2761 015146  111100                 20$:MOVB    (R1),R0                 ;GET EXPD BYTE
2762 015150  042700  177400             BIC     #^C<377>,R0             ;CLEAR UPPER BYTE
2763 015154  110037  015470             MOVB    R0,PRBEXP               ;SAVE FOR ERROR REPORT
2764 015160  111203                     MOVB    (R2),R3                 ;GET RECV BYTE
2765 015162  042703  177400             BIC     #^C<377>,R3             ;CLEAR UPPER BYTE
2766 015166  110337  015472             MOVB    R3,PRBREC               ;FOR ERROR REPORT
2767 015172                             XOR     R0,R3                   ;XOR EXPD/RECV
2768 015202  122122                     CMPB    (R1)+,(R2)+             ;EXPD = RECV?
2769 015204  001431                     BEQ     30$                     ;BR IF YES
2770 015206  005237  002320             INC     PRMNO                   ;UPDATE ERROR COUNT
2771 015212  023727  002320  000010     CMP     PRMNO,#8.               ;PRINTED 8?
2772 015220  101023                     BHI     30$                     ;BR IF YES
2773 015222                         27$:PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
     015222  010346                     MOV     R3,-(SP)
     015224  013746  015472             MOV     PRBREC,-(SP)
     015230  013746  015470             MOV     PRBEXP,-(SP)
     015234  010446                     MOV     R4,-(SP)
     015236  012746  015336             MOV     #PRBMSG,-(SP)
     015242  012746  000005             MOV     #5,-(SP)
     015246  010600                     MOV     SP,R0
     015250  104415                     TRAP    C$PNTX
     015252  062706  000014             ADD     #14,SP
2774 015256                             FORCEXIT        50$             ;@@D
2775 015266  000404                     BR      35$                     ;@@D
2776 015270                         30$:
2777 015270                             FORCERROR  27$,NOTSSR           ;@@D
2778 015300                         35$:                                ;@@D
2779 015300  005204                     INC     R4                      ;NUMBER OF THE NEXT
2780 015302  020405                     CMP     R4,R5                   ;DONE ALL YET?
2781 015304  002001                     BGE     50$                     ;BR IF YES
2782 015306  000717                     BR      20$                     ;DO ANOTHER
2783 015310                         50$:PRINTX  #PRBTOT,PRMNO           ;PRINT TOTAL ERROR COUNT
     015310  013746  002320             MOV     PRMNO,-(SP)
     015314  012746  015423             MOV     #PRBTOT,-(SP)
     015320  012746  000002             MOV     #2,-(SP)
     015324  010600                     MOV     SP,R0
```

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```
        015326  104415                          TRAP    C$PNTX
        015330  062706  000006                  ADD     #6,SP
   2784 015334  000207                          RTS     PC              ;RETURN
   2785
   2786 015336     045     116     045  PRBMSG: .ASCIZ  '<N><A>   BYTE #<D2><A>  EXPD: <03><A>  RECV: <03><A>  XOR: <03>'
   2787 015423     045     116     045  PRBTOT: .ASCIZ  '<N><A> NUMBER OF BYTES IN ERROR = <D2>'
   2788                                          .EVEN
   2789 015470  000000                  PRBEXP: .WORD   0               ;EXPD
   2790 015472  000000                  PRBREC: .WORD   0               ;RECV
   2791                                          .SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
   2792                                  ;+
   2793                                  ;
   2794                                  ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
   2795                                  ;
   2796                                  ;INPUTS:
   2797                                  ;
   2798                                  ;       R1      RECEIVED DATA
   2799                                  ;       R2      EXPECTED DATA
   2800                                  ;
   2801                                  ;-
   2802
   2803 015474                          EXPREC::BGNMSG  EXPREC
        015474
   2804 015474  004737  007742                  JSR     PC,PRIXOR               ;PRINT THE DATA
   2805 015500                                  ENDMSG
        015500
        015500  104423                  L10017: TRAP    C$MSG
   2806                                          .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
   2807                                  ;+
   2808                                  ;
   2809                                  ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
   2810                                  ;
   2811                                  ;INPUTS:
   2812                                  ;
   2813                                  ;       R1      RECEIVED DATA BYTE
   2814                                  ;       R2      EXPECTED DATA BYTE
   2815                                  ;
   2816                                  ;-
   2817
   2818
   2819 015502                          EXPBREC::BGNMSG EXPBREC
        015502
   2820 015502  004737  007612                  JSR     PC,PRIBXOR              ;PRINT THE DATA
   2821 015506                                  ENDMSG
        015506
        015506  104423                  L10020: TRAP    C$MSG
   2822
   2823
   2824                                          .SBTTL  RAMERR  - PRINT RAM AND PACKET DATA
   2825                                  ;+
   2826                                  ;
   2827                                  ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
   2828                                  ;
   2829                                  ;INPUTS:
   2830                                  ;
   2831                                  ;       R4      POINTER TO COMMAND PACKET
   2832                                  ;
```

RAMERR - PRINT RAM AND PACKET DATA

```
                                        ;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
                                        ;-
                                        
015510                                  BGNMSG  RAMERR
015510                                  RAMERR::
015510  004737  013776                  JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
015514                                  ENDMSG
015514                                  L10021:
015514  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
                                        ;-
                                        
015516                                  BGNMSG  RAMTADD
015516                                  RAMTADD::
015516  004737  010274                  JSR     PC,PRITADD      ;PRINT TEST ADDRESS
015522  004737  013776                  JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
015526                                  ENDMSG
015526                                  L10022:
015526  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
```

RAMEXP - PRINT RAM EXPD/RECV DATA

```
                                        ;       R4      CONTROLLER RAM ADDRESS
                                        ;-
                                        ;
                                                BGNMSG  RAMEXP
                                        RAMEXP::
015530  042701  177400                          BIC     #^C<377>,R1             ;SAVE EXPD RAM DATA BYTE
015534  042702  177400                          BIC     #^C<377>,R2             ;SAVE EXPD RAM DATA BYTE
015540  004737  010066                          JSR     PC,PRIRAM               ;PRINT THE RAM ADDRESS
015544  004737  007742                          JSR     PC,PRIXOR               ;PRINT THE DATA
015550                                          ENDMSG
                                        L10023:
015550  104423                                  TRAP    C$MSG
                                                .SBTTL  TIMEXP  - PRINT TIMER A,B AND EXP/REC
                                        ;*
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                        ;AND TIMER A,B HEADER MESSAGE
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      RECEIVED DATA
                                        ;       R2      EXPECTED DATA
                                        ;-
                                        ;
                                                BGNMSG  TIMEXP
                                        TIMEXP::
015552                                          PRINTX  #TIMSGO                 ;PRINT HEADER
015552  012746  015600                          MOV     #TIMSGO,-(SP)
015556  012746  000001                          MOV     #1,-(SP)
015562  010600                                  MOV     SP,R0
015564  104415                                  TRAP    C$PNTX
015566  062706  000004                          ADD     #4,SP
015572  004737  007742                          JSR     PC,PRIXOR               ;PRINT THE DATA
015576                                          ENDMSG
                                        L10024:
015576  104423                                  TRAP    C$MSG

015600      045     116     045 TIMSGO: .ASCIZ  '<N><A TIMER A STATUS IS IN BIT 3<N><A TIMER B STATUS IS IN BIT 2'
                                        .EVEN
                                                .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

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

                                                BGNMSG  BADSSR
                                        BADSSR::
015700                                          MOV     R2,-(SP)                ;SAVE DATA TRANSFERRED
015702  042702  177400                          BIC     #177400,R2              ;GET JUST ONE BYTE
```

BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

```
   2929 015706                                  PRINTB  #XFERASC,R2
        015706  010246                          MOV     R2,-(SP)
        015710  012746  015740                  MOV     #XFERASC,-(SP)
        015714  012746  000002                  MOV     #2,-(SP)
        015720  010600                          MOV     SP,R0
        015722  104414                          TRAP    C$PNTB
        015724  062706  000006                  ADD     #6,SP
   2930 015730  012602                          MOV     (SP)+,R2                ;RESTORE R2
   2931 015732  004737  006024                  JSR     PC,PRITSSR              ;DECODE TSSR CONTENTS
   2932 015736                                  ENDMSG
        015736                          L10025:
        015736  104423                          TRAP    C$MSG
   2933 015740     045     116     045  XFERASC:        .ASCIZ  '%N%A Data Transferred = %03'
```

GLOBAL SUBROUTINES SECTION

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

                                ;++
                                ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
                                ; THAT ARE USED IN MORE THAN ONE TEST.
                                ;--
                                        .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER

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

                                SOFINIT::
                                        SAVREG                          ; SAVE THE REGISTERS
016000  012765  000000  000002          MOV     #0,TSSR(R5)             ; DO THE INIT.
016006  004737  016250                  JSR     PC,WAITF                ; WAIT FOR SSR
016012  016500  000002                  MOV     TSSR(R5),R0             ;GET THE TSSR REGISTER
016016  010004                          MOV     R0,R4                   ;TSSR CONTENTS
016020  042704  176277                  BIC     #^C<HIADDR!OFL>,R4
016024  052704  002200                  BIS     #SSR!NBA,R4             ;R4 HAS EXPECTED CONTENTS
016030  020400                          CMP     R4,R0                   ;ONLY EXPECTED BITS SET ?
016032  001402                          BEQ     5$                      ;BRANCH IF OKAY
016034  000241                          CLC                             ;CLEAR THE CARRY FOR ERROR
016036  000401                          BR      10$                     ;GO TO EXIT
016040  000261                  5$:     SEC                             ;SET THE CARRY BIT
016042  000207                  10$:    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
                                ;
                                ;-

016044                          CHKAMB:
016044                                  SAVREG                  -   ;SAVE THE GENERAL REGISTERS
016050  010004                          MOV     R0,R4               ;CONTENTS OF TSSR
016052  032700  100000                  BIT     #SC,R0              ;IS BIT 15 SET ?
016056  001004                          BNE     5$                  ;BRANCH IF YES
016060  032700  174077                  BIT     #^C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
016064  001023                          BNE     40$                 ;MUST BE AN ERROR
016066  000424                          BR      45$                 ;RETURN WITH SUCCESS
016070  032700  000200          5$:     BIT     #SSR,R0             ;IS READY BIT SET ?
016074  001011                          BNE     10$                 ;BRANCH IF READY BIT IS SET.
016076  032700  000040                  BIT     #BIT5,R0            ;IS FATAL ERROR BIT SET ?
016102  001414                          BEQ     40$                 ;ERROR IF NOT
016104  042704  177761                  BIC     #^CTERCLS,R4        ;CLEAR ALL BUT TERMINATION CODE
016110  020427  000016                  CMP     R4,#16              ;ALL THREE BITS MUST BE SET
016114  001007                          BNE     40$                 ;ERROR IF NOT SET
016116  000410                          BR      45$                 ;OK IF ALL ARE SET
016120  032700  000040          10$:    BIT     #BIT5,R0            ;IS FATAL ERROR BIT SET ?
016124  001405                          BEQ     45$                 ;ERROR IF BIT IS SET WITH SSR
016126  032700  000006                  BIT     #BIT2!BIT1,R0       ;IS THIS A FUNCTION REJECT
016132  001002                          BNE     45$                 ;BR, IF TSSR IS OK
016134  000241                  40$:    CLC                         ;AMBIGUOUS CONTENTS
016136  000401                          BR      50$
016140  000261                  45$:    SEC                         ;SHOW SUCCESS - NO AMBIGUITY
016142  000207                  50$:    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
016144     000                  INTMASK:        .BYTE   0
                                ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
016145     000                  INTFLAG:        .BYTE   0

                                ;SAVED INTERRUPT VECTOR:
016146  C00000                  INTVEC: .WORD   0
                                ;SAVE CPU PC
016150  000000                  INTCPC: .WORD   0

                                ;SUBROUTINE TO ENABLE INTERRUPTS:
016152  010046                  ENAINT: MOV     R0,-(SP)        ;SAVE R0
016154  013700  002210                  MOV     IVEC,R0         ;GET POINTER TO VECTORS
016160  012720  016216                  MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
016164  012720  000340                  MOV     #PRI07,(R0)+
016170  012600                          MOV     (SP)+,R0        ;RESTORE R0
016172  011646                          MOV     (SP),-(SP)
016174  012766  000000  000002          MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
016202  000002                          RTI

                                ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
016204  011646                  DSBINT: MOV     (SP),-(SP)
016206  012766  000340  000002          MOV     #PRI07,2(SP)
016214  000002                          RTI
                                        .SBTTL  INTR    - INTERRUPT HANDLERS

                                        BGNSRV  INTR            ;DEFINE INTERRUPT ENTRY
016216
016216                          INTR::
016216  012737  000001  002224          MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
016224  105037  016145                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
016230  132737  000001  016144          BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
016236  001003                          BNE     1$              ;BR IF YES
016240  152737  000001  016145          BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.

                                ;SAVE REGISTERS, MSG BUFFER, ETC.
016246                          1$:
016246                                  ENDSRV
016246                          L10026:
016246  000002                          RTI
```

WAITF - WAIT FOR SUBSYSTEM READY

```
                                        .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
                                ;
016250  000401                  WAITF:: BR      1$              ;NOP WHEN SUPER FIXED
016252                                  BREAK                   ; DO A SUPVSR BREAK FIRST.
016252  104422                          TRAP    C$BRK
016254  012746  011000          1$:     MOV     #11000,-(SP)    ;25-APRIL-83 REV 8  - 1100 MSEC TIMER
016260  C16500  000002          2$:     MOV     TSSR(R5),R0     ;READ THE TSSR REGISTER
016264  105700                          TSTB    R0              ;TEST FOR READY BIT SET
                                ;
016266  100420                          BMI     3$              ; EXIT ON STOP FLAG.
016270                                  DELAY   1               ; WAIT 100 USEC
016270  012727  000001                  MOV     #1,(PC)+
016274  000000                          .WORD   0
016276  013727  002116                  MOV     L$DLY,(PC)+
016302  000000                          .WORD   0
016304  005367  177772                  DEC     -6(PC)
016310  001375                          BNE     .-4
016312  005367  177756                  DEC     -22(PC)
016316  001367                          BNE     .-20
016320  005316                          DEC     (SP)            ;REDUCE DELAY COUNT
016322  001356                          BNE     2$              ;RETRY UNTIL TIMER EXPIRES
016324  000241                          CLC                     ; C = 0, CONTROLLER STILL RUNNING...
016326  000401                          BR      4$              ;...OR HUNG-UP AFTER 300 MSEC.
016330  000261                  3$:     SEC                     ; C = 1, CONTROLLER IS STOPPED.
016332  005326                  4$:     DEC     (SP)+           ;RESTORE STACK WITHOUT CHANGING CARRY BIT
016334  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
                                        ;-
016336                                  CHKTSSR:
016336  004737  016250                          JSR     PC,WAITF                ;WAIT FOR READY
016342  103014                                 BCC     20$                     ;BRANCH IF TIME OUT
016344  004737  016044                          JSR     PC,CHKAMB               ;TSSR AMBIGOUS?
016350  103006                                 BCC     10$                     ;BR IF YES
016352  C32700  100000                         BIT     #SC,R0                  ;SPECIAL CONDITION SET?
016356  001405                                 BEQ     15$                     ;BR IF NO
016360  032700  074000                         BIT     #<SCE!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
016364  001402                                 BEQ     15$                     ;BR IF NO
016366  000241                          10$:    CLC                             ;SET FAILURE
016370  000401                                 BR      20$                     ;
016372  000261                          15$:    SEC                             ;SET SUCCESS
016374  000207                          20$:    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.
016376  012737  016430  000004          XNXM:   MOV     #2$,@#4         ; SET BUSERR VECTOR.
016404  012737  000200  000006                  MOV     #PRI04,@#6
016412  005003                                 CLR     R3                      ;FLAG.
016414  005711                          1$:    TST     (R1)                    ;TEST THE ADDRESS(ES).
                                                                               ;IF ANY TRAP, CONTINUE AT 2$.
016416  020102                                 CMP     R1,R2                   ;OTHERWISE, CONTINUE HERE.
016420  001407                                 BEQ     3$                      ;BR IF FINISHED (NO NEXM'S).
016422  062701  000002                         ADD     #2,R1                   ;SET NEXT ADDRESS...
016426  000772                                 BR      1$                      ;...AND CONTINUE.
016430  005103                          2$:    COM     R3                      ;GOT ONE, SET FLAG...
016432  012716  016440                         MOV     #3$,(SP)
016436  000002                                 RTI                             ;...AND DISMISS INTERRUPT...
016440                                  3$:    CLRVEC  #4                      ;...AND GIVE BACK THE VECTOR.
016440  012700  000004                         MOV     #4,R0
016444  104436                                 .TRAP   C$CVEC
016446  005703                                 TST     R3                      ;DID WE CATCH ONE ??
016450  001401                                 BEQ     .+4                     ;NO, "C" = 0, SKIP NEXT.
016452  000261                                 SEC                             ;YES, "C" = 1, (R1) = NEXM ADDR.
016454  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
                                ;
016456                          TSTLOOP::
016456  005737  002170                  TST     NOITS           ; ITERATIONS INHIBITED?
016462  001006                          BNE     1$              ; YES.
016464  005737  002204                  TST     QVP             ; NO.
016470  100403                          BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
016472  005337  002216                  DEC     LOOPCNT         ; BUMP LOOP COUNTER.
016476  001002                          BNE     2$
016500  000241                  1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
016502  000401                          BR      3$
016504  000261                  2$:     SEC                     ;LOOP ENABLED.
016506  000207                  3$:     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
                                ;
                                ;-

016510                          TSTSETUP::
016510  010046                          MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
016512  005037  003154                  CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
016516  005037  016756                  CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
016522  005037  005772                  CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
016526  105037  016144                  CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
016532  013700  002202                  MOV     UNITN,R0        ; GET THE UNIT NUMBER,
016536  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
016540  005737  003114                  TST     NODEV           ; DID STARTUP FIND THE DEVICE?
016544  001430                          BEQ     4$              ; BR IF YES
016546  100010                          BPL     3$              ; BR IF NOT IDLE
```

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```
3219 016550  052760  160000  003176          BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
3220 016556                                  ERRDF   1,NXR,NXRERR    ; NO DEVICE HERE -- PRINT IT
     016556  104455                          TRAP    C$ERDF
     016560  000001                          .WORD   1
     016562  003740                          .WORD   NXR
     016564  005736                          .WORD   NXRERR
3221 016566  000407                          BR      2$
3222 016570  052760  160001  003176  3$:     BIS     #160001,ERTABL(R0)      ; FLAG ERROR IN THE ERROR TABLE
3223 016576                                  ERRDF   2,NOINIT        ; DEVICE NOT IDLE
     016576  104455                          TRAP    C$ERDF
     016600  000002                          .WORD   2
     016602  004335                          .WORD   NOINIT
     016604  000000                          .WORD   0
3224 016606  012737  177777  003112  2$:     MOV     #-1,DUFLG       ; DROP THE UNIT
3225 016614                                  DODU    UNITN
     016614  013700  002202                  MOV     UNITN,R0
     016620  104451                          TRAP    C$DODU
3226 016622                                  DOCLN                   ; ABORT THE PASS
     016622  104444                          TRAP    C$DCLN
3227 016624  000423                          BR      5$
3228
3229 016626                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
     016626  104421                          TRAP    C$RFLA
3230 016630  032700  001000                  BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
3231 016634  001412                          BEQ     1$              ; BR IF NO
3232 016636  011600                          MOV     (SP),R0         ;GET THE ID MESSAGE
3233 016640                                  PRINTF  #TNAM,R0                ;DISPLAY THE TEST ID
     016640  010046                          MOV     R0,-(SP)
     016642  012746  016704                  MOV     #TNAM,-(SP)
     016646  012746  000002                  MOV     #2,-(SP)
     016652  010600                          MOV     SP,R0
     016654  104417                          TRAP    C$PNTF
     016656  062706  000006                  ADD     #6,SP
3234 016662  005237  002214          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
3235 016666                                  SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
     016666  013700  002212                  MOV     IPRI,R0
     016672  104441                          TRAP    C$SPRI
3236 016674  005726                  5$:     TST     (SP)+           ;FIX UP THE STACK
3237 016676  013705  002206                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
3238 016702  000207                          RTS     PC
3239 016704     045     123     045  TNAM:   .ASCIZ  '%S%T%A Test'
3240                                          .EVEN
3241                                          .SBTTL  TSTEND  - PRINT ERRORS RECEIVED
3242                                  ;
3243                                  ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
3244                                  ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
3245                                  ;
3246 016720                          TSTEND: RFLAGS  R0
     016720  104421                          TRAP    C$RFLA
3247 016722  030027  020000                  BIT     R0,#IER
3248 016726  001412                          BEQ     1$              ; BR IF "IER" NOT SET.
3249 016730                                  PRINTF  #ESUM,ERRK      ; PRINT ERROR COUNT.
     016730  013746  016756                  MOV     ERRK,-(SP)
     016734  012746  016760                  MOV     #ESUM,-(SP)
     016740  012746  000002                  MOV     #2,-(SP)
     016744  010600                          MOV     SP,R0
     016746  104417                          TRAP    C$PNTF
```

TSTEND - PRINT ERRORS RECEIVED

```
        016750  062706  000006                  ^JD     #6,SP
   3250 016754  000207              1$:        RTS     PC
   3251
   3252 016756  000000              ERRK:      0                       ; LOCAL ERROR COUNT.
   3253 016760     045     101      040 ESUM:   .ASCIZ  /%A %D%A ERRORS/
   3254 016777     105     122      122 EMAXDU: .ASCIZ  /ERROR LIMIT REACHED    DROPPING UNIT/
   3255                                        .EVEN
```

```
                                        .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
                                ;+
                                ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
                                ;-
017044  005237  016756          INCERK: INC     ERRK            ; INCREMENT LOCAL ERROR COUNT
017050  010046                  MOV     R0,-(SP)        ; SAVE R0
017052  013700  002202                  MOV     UNITN,R0        ; GET UNIT NUMBER,
017056  006300                  ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
017060  062700  003176                  ADD     #ERTABL,R0      ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
017064  005210                  INC     (R0)            ; INCREMENT THE DEVICE ERROR COUNT
017066  032710  007777                  BIT     #7777,(R0)      ; DID WE OVERFLOW THE FIELD?
017072  001001                  BNE     1$              ; BR IF NO.
017074  005310                  DEC     (R0)            ; YES -- BACK IT UP TO 7777.
017076  012600                  1$:     MOV     (SP)+,R0        ; RESTORE R0
017100  000207                  RTS     PC              ; RETURN TO CALLER

017102  010046                  CKEMAX: MOV     R0,-(SP)        ; SAVE R0
017104  013700  002202                  MOV     UNITN,R0        ; GET UNIT NUMBER
017110  C06300                  ASL     R0              ; ... AND MAKE IT A WORD OFFSET
017112  016000  003176                  MOV     ERTABL(R0),R0   ; GET ERROR TABLE ENTRY
017116  042700  170000                  BIC     #170000,R0      ; EXTRACT ERROR COUNT FIELD
017122  020037  002174                  CMP     R0,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
017126  103004                  BHIS    1$              ; BR IF YES
017130  023737  016756  002172        CMP     ERRK,LERRMAX    ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
017136  103417                  BLO     2$              ; BR IF NO
017140                          1$:     RFLAGS  R0              ; GET OPERATOR FLAGS
017140  104421                  TRAP    C$RFLA
017142  032700  000040                  BIT     #IDU,R0         ; IS DROPPING INHIBITED?
017146  001013                  BNE     2$              ; BR IF YES.
017150  012737  177777  003112        MOV     #-1,DUFLG       ; NO -- DROP THE UNIT
017156                          ERRDF   4,EMAXDU
017156  104455                  TRAP    C$ERDF
017160  000004                  .WORD   4
017162  016777                  .WORD   EMAXDU
017164  000000                  .WORD   0
017166                          DODU    UNITN
017166  013700  002202          MOV     UNITN,R0
017172  104451                  TRAP    C$DODU
017174                          DOCLN
017174  104444                  TRAP    C$DCLN
017176  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
017200  000207                  RTS     PC              ; RETURN TO CALLER
```

```
                                        .SBTTL  CKDROP  - CHECK IF UNIT SHOULD BE DROPPED
                                ;+
                                ; CHECK IF UNIT SHOULD BE DROPPED
                                ;-
017202  010046                  CKDROP: MOV     R0,-(SP)
017204                                  FORCERROR       1$,NOTSSR
017214                                  RFLAGS  R0
017214  104421                          TRAP    C$RFLA
017216  032700  000040                  BIT     #IDU,R0
017222  001010                          BNE     1$
017224  011600                          MOV     (SP),R0
017226  012737  177777  003112          MOV     #-1,DUFLG
017234                                  DODU    UNITN
017234  013700  002202                  MOV     UNITN,R0
017240  104451                          TRAP    C$DODU
017242                                  DOCLN                   ;ABORT THE PASS
017242  104444                          TRAP    C$DCLN
017244  012600                  1$:     MOV     (SP)+,R0
017246  000207                          RTS     PC

                                        .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
                                ;
                                ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
                                ;
017250                          CONFIG:
017250  004737  015774                  JSR     PC,SOFINIT
017254  000207                          RTS     PC
                                        .SBTTL  KTON,KTOFF      - ENABLE/DISABLE MEMORY MANAGEMENT
                                ;
                                ; SUBROUTINE - ENABLE MEM MGT.
                                ;
017256  005737  003132          KTON:   TST     KTFLG           ; GOT KT?
017262  001403                          BEQ     1$              ; NO.
017264  012737  000001  177572          MOV     #1,SR0          ; YES. ENABLE KT11.
017272  000207                  1$:     RTS     PC

                                ;
                                ; SUBROUTINE - DISABLE MEM MGT.
                                ;
017274  005737  003132          KTOFF:  TST     KTFLG           ; GOT KT11?
017300  001405                          BEQ     1$              ; NO.
017302  000240                          NOP
017304  000240                          NOP
017306  012737  000000  177572          MOV     #0,SR0          ; DISABLE KT.
017314  000207                  1$:     RTS     PC
```

```
                                        .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
                                ;-
017316                          SETMAP:
017316                                  SAVREG                          ;SAVE R1-R4 UNTIL NEXT RETURN
017322  005737  003132                  TST     KTFLG                   ;SYSTEM HAVE ABOVE 28K?
017326  001433                          BEQ     10$                     ;BR IF NO
017330  010102                          MOV     R1,R2                   ;SAVE LOW ORDER BITS
        000006                          .REPT   6
                                        ASR     R0                      ;CONVERT WORD ADDRESS TO 32W BLOCKS
                                        ROR     R1                      ;MAKE IT DOUBLE PRECISION
                                        .ENDR
017362  042701  000177                  BIC     #177,R1                 ;ALIGN FOR LOWER 4K BOUNDARY
017366  020137  003132                  CMP     R1,KTFLG                ;HIGHER THAN EXISTING MEMORY?
017372  103011                          BHIS    10$                     ;BR IF YES
017374  010137  172352                  MOV     R1,@#KIPAR5             ;SETUP MAPPING REGISTER PAR5
017400  042702  160000                  BIC     #160000,R2              ;SETUP DISPLACEMENT IN PAGE
017404  062702  120000                  ADD     #120000,R2              ;ADD IN PAR5 BIAS
017410  010200                          MOV     R2,R0                   ;RETURN IN R0
017412  000261                          SEC                             ;SET SUCCESS
017414  000401                          BR      15$                     ;
017416  000241                  10$:    CLC                             ;SET FAILURE
017420  000207                  15$:    RTS     PC                      ;RETURN
                                        .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
                                ;
```

REGSAV - SAVE R1-R5 ON STACK

```
3392                                    ;-
3393
3394 017422                             REGSAV:
3395 017422  010446                             MOV     R4,-(SP)
3396 017424  010346                             MOV     R3,-(SP)
3397 017426  010246                             MOV     R2,-(SP)
3398 017430  010146                             MOV     R1,-(SP)
3399 017432  010546                             MOV     R5,-(SP)
3400 017434  016605  000012                     MOV     10.(SP),R5
3401 017440  004736                             JSR     PC,@(SP)+
3402 017442  012601                             MOV     (SP)+,R1
3403 017444  012602                             MOV     (SP)+,R2
3404 017446  012603                             MOV     (SP)+,R3
3405 017450  012604                             MOV     (SP)+,R4
3406 017452  012605                             MOV     (SP)+,R5
3407 017454  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
                                ;-
017456                          GETPAT::
                                        SAVREG                          ;SAVE THE GENERAL REGISTERS
017462                          1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
017462  104443                          TRAP    C$GMAN
017464  000406                          BR      10000$
017466  017512                          .WORD   PATDAT
017470  C00022                          .WORD   T$CODE
017472  017514                          .WORD   DATASC
017474  000377                          .WORD   377
017476  000000                          .WORD   T$LOLIM
017500  000377                          .WORD   T$HILIM
017502                          10000$:
017502                                  BNCOMPLETE      1$      ;RETRY IF ERROR
017502  103367                          BCC     1$
017504  013700  017512                  MOV     PATDAT,R0       ;DATA PATTERN FROM OPERATOR
017510  000207                          RTS     PC              ;RETURN TO CALLER

                                ;+
                                ;LOCAL DATA AREA
                                ;-

017512  000000                  PATDAT: .WORD   0               ;TEMPORARY STORAGE FOR DATA
017514     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
                                ;-
017540                          GETSEL::
017540                                  SAVREG                          ;SAVE GENERAL REGISTERS
017544  010002                          MOV     R0,R2                   ;SAVE THE MENU ADDRESS
017546  010203                  1$:     MOV     R2,R3                   ;START OF MENU STRING
017550  005713                  2$:     TST     (R3)                    ;END OF ASCII ?
017552  001412                          BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
017554                                  PRINTF  @SELASC,(R3)+           ;DISPLAY THE MENU
017554  C12346                          MOV     (R3)+,-(SP)
017556  012746  017724                  MOV     @SELASC,-(SP)
017562  012746  000002                  MOV     @2,-(SP)
017566  010600                          MOV     SP,R0
017570  104417                          TRAP    C$PNTF
017572  062706  000006                  ADD     @6,SP
017576  000764                          BR      2$
017600                          3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
017600  104443                          TRAP    C$GMAN
017602  000406                          BR      10001$
017604  017760                          .WORD   MENRES
017606  000042                          .WORD   T$CODE
017610  017731                          .WORD   MENASC
017612  177777                          .WORD   -1
017614  000000                          .WORD   T$LOLIM
017616  177777                          .WORD   T$HILIM
017620                          10001$:
017620                                  BNCOMPLETE      1$              ;RETRY IF ERROR
017620  103352                          BCC     1$
017622  013700  017760                  MOV     MENRES,R0               ;GET THE OPERATOR'S REPLY
017626  020001                          CMP     R0,R1                   ;COMPARF TO MAXIMUM ALLOWED
017630  101411                          BLOS    5$                      ;BRANCH IF OK
017632                                  PRINTF  @MENERR                 ;DISPLAY ERROR MESSAGE
017632  012746  017656                  MOV     @MENERR,-(SP)
017636  012746  000001                  MOV     @1,-(SP)
017642  010600                          MOV     SP,R0
017644  104417                          TRAP    C$PNTF
017646  062706  000004                  ADD     @4,SP
017652  000735                          BR      1$                      ;RETRY
017654  000207                  5$:     RTS     PC                      ;RETURN TO CALLER
017656     045     116     045  MENERR: .ASCIZ  '%N%A *** Menu Selection Too Large ***'
017724     045     116     045  SELASC: .ASCIZ  '%N%T'
017731     105     156     164  MENASC: .ASCIZ  'Enter Menu Selection: '
                                        .EVEN
017760  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.
                                ;
                                ;-

017762                          CHKMAN::
017762                                  SAVREG                  ;SAVE THE REGISTERS
017766                                  MANUAL                  ;SEE IF MANUAL INTERVENTION OK
017766  104450                          TRAP    C$MANI
017770                                  BCOMPLETE 1$            ;BRANCH IF ALLOWED
017770  103411                          BCS     1$
017772                                  PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
017772  012746  020016                  MOV     #NOMAN,-(SP)
017776  012746  000001                  MOV     #1,-(SP)
020002  010600                          MOV     SP,R0
020004  104417                          TRAP    C$PNTF
020006  062706  000004                  ADD     #4,SP
020012  000241                          CLC                     ;CLEAR CARRY FOR ERROR
020014  000207                  1$:     RTS     PC              ;RETURN

020016      045     116     045 NOMAN:  .ASCIZ  '%N%A *** Manual Intervention not Allowed - Test Aborted ***'
                                        .even
```

```
                                        .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
                                ;
                                ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
                                ;
020112                          ENVIRN: MEMORY  R0
020112  104431                          TRAP    C$MEM
020114  010037  003124                  MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
020120  062737  000002  003124          ADD     #2,FREE
020126  011037  003126                  MOV     (R0),FRESIZ     ;...AND WORD COUNT.
020132  162737  000004  003126          SUB     #4,FRESIZ
020140  013702  002012                  MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
020144  162737  000007  003126  10$:    SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
020152  005302                          DEC     R2
020154  001373                          BNE     10$
020156  013700  003124                  MOV     FREE,R0         ;GET FIRST FREE ADDRESS
020162  063700  003126                  ADD     FRESIZ,R0       ;POINT TO LAST FREE ADDRESS
020166  162700  000002                  SUB     #2,R0           ;BACKUP 1 WORD
020172  010037  003130                  MOV     R0,FREEHI       ;STORE LAST FREE ADDRESS
020176  000240                          NOP                     ;*****************************************
020200  012701  177520                  MOV     #BDVPCR,R1      ;GET BDV11 PCR ADDRESS
020204  010102                          MOV     R1,R2           ;COPY TO R2
020206  062702  000002                  ADD     #2,R2           ;SET THE RANGE
020212  004737  016376                  JSR     PC,XNXM         ;SEE IF WE HAVE ONE
020216  103001                          BCC     15$             ;OK TO SET FLAGS
020220  000423                          BR      40$             ;RETURN WITH FLAGS CLEAR
020222  013701  177520          15$:    MOV     BDVPCR,R1       ;SAVE PCR CONTENTS
020226  062701  000001                  ADD     #1,R1           ;ADD ONE TO IT
020232  012702  177520                  MOV     #BDVPCR,R2      ;GET BDV11 PCR ADDRESS
020236  005212                          INC     (R2)            ;TRY TO WRITE TO IT
020240  013703  177520                  MOV     BDVPCR,R3       ;GET RESULTS
020244  020103                          CMP     R1,R3           ;DID IT CHANGE?
020246  001006                          BNE     20$             ;NO, MUST BE 11/23B
020250  005237  003144                  INC     T23A            ;SET THE FLAG
020254  042737  170000  002120          BIC     #170000,L$HIME  ;SUPERVISOR COULD BE WRONG
                                ;               NOP             ;BR 40$ FOR RELEASE
                                ;               PRINTF  #M8186  ;TELL THE SYSTEM TYPE
020262  000402                          BR      40$             ;RETURN
020264  005237  003146          20$:    INC     T23B            ;SET THE FLAG
                                ;               NOP             ;BR 40$ FOR RELEASE
                                ;               PRINTF  #M8189  ;TELL THE SYSTEM TYPE
020270                          40$:
020270  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
                                ;+
                                ;
                                ;ROUTINE TO INIT KT-11
                                ;
                                ;-
                                KTINIT:
020272                                  CLR     KTFLG           ; INIT >28K MEMORY FLAG
020272  005037  003132                  CLR     KTENABLE        ; INIT TEST >28K FLAG
020276  005037  003134                  CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
020302  023727  002120  001577          BLOS    9$              ; NO.
020310  101454                          MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
020312  013700  000004                  MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
020316  012737  020430  000004          TST     @#SR0           ; GOT KT11?
020324  005737  177572                  NOP                     ;  (TRAP IF NO).
020330  000240                          MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
020332  013737  002120  003132          CMP     #10000,KTFLG    ; MORE THAN 18 BITS?
020340  022737  010000  003132          BMI     4$              ; NO
020346  100404                          BIC     #3777,KTFLG     ;
020350  042737  003777  003132          BR      5$              ;
020356  000403                  4$:     BIC     #177,KTFLG      ;
020360  042737  000177  003132  5$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
020366  010037  000004                  CLR     R0              ; R0 = AR DATA.
020372  005000                          MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
020374  012701  172340                  MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
020400  012761  077406  177740  1$:     MOV     R0,(R1)+        ; SET KIPAR REG.
020406  010021                          ADD     #200,R0         ; BUMP AR DATA BY "4K".
020410  062700  000200                  CMP     R0,#2000        ; AT "I/O"?
020414  020027  002000                  BNE     1$              ; NO.
020420  001367                          MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
020422  012741  177600                  BR      9$
020426  000405
                                2$:     MOV     #6$,(SP)        ; SET UP RETURN
020430  012716  020436                  RTI                     ; RTI TO NEXT LOCATION
020434  000002
                                6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
020436  010037  000004
                                9$:     RTS     PC
020442  000207
```

```
                                        ;*
                                        ;       SUBROUTINE TO SET EXTENDED FEATURES SWITCH
                                        ;
                                        ;       Requires that SOFINIT and WRTCHR have been done previous to call.
                                        ;
                                        ;
                                        ;INPUTS:
                                        ;       R5      CURRENT UNIT NUMBER
                                        ;OUTPUTS:
                                        ;       The Extended Features Switch is set.
                                        ;
                                        ;-

020444                                  INVERT::
020444  013705  002206                          mov     ceraddr,r5
020450  004737  015774                          jsr     pc,sofinit
020454  103406                                  bcs     25$
020456  010001                                  mov     r0,r1
020460                                          errdf   errno,sfierr,sfimsg
020460  104455                                  TRAP    C$ERDF
020462  002261                                  .WORD   1201
020464  003652                                  .WORD   sfierr
020466  012034                                  .WORD   sfimsg
020470  104455                                                          trap c$erdf
020472                                  25$:    ckloop
020472  104406                                  TRAP    C$CLP1
020474  013737  002202  076300                  mov     unitn,t39dsw
020502  012704  076260                          mov     #t39pk2,r4
020506  004737  010662                          jsr     pc,wrtchr
020512  103406                                  bcs     50$
020514  010001                                  mov     r0,r1
020516                                          errhrd  errno,wrtmsg,sfimsg
020516  104456                                  TRAP    C$ERHRD
020520  002261                                  .WORD   12C1
020522  005056                                  .WORD   wrtmsg
020524  012034                                  .WORD   sfimsg
020526  104456                                                          trap c$erhrd
020530                                  50$:    ckloop
020530  104406                                  TRAP    C$CLP1
020532  013701  075610                          mov     t39bfr+12,r1
020536  032701  000200                          bit     #bit7,r1
020542  001020                                  bne     1$
020544  012737  100206  020610          10$:    MOV     #100206,CMDPKT          ; WRT SUB-SYS MEM CMD
020552  012737  020620  020612                  MOV     #WSMBK,CMDPKT+2         ; MSG BUF ADDR
020560  012737  000006  020616                  MOV     #6,CMDPKT+6             ; BYTE COUNT
020566  012737  100010  020620                  MOV     #100010,WSMBK           ; INVERT THE SWITCH
020574  012704  020610                          MOV     #CMDPKT,R4              ; SET CMDPKT INTO R4
020600  004737  010662                          JSR     PC,WRTCHR               ; DO IT
020604  000207                          1$:     RTS     PC                      ; RETURN

                                        ;       COMMAND PACKET.

020610                                  .       =       <.+3>&177774    ;MUST BE ON MOD 4 BOUNDRY.

020610  000000                          CMDPKT:: 0                      ;1ST WORD IS TSO5 COMMAND.
020612  000000                                    0                     ;2ND WORD IS THE BUFFER LOW ADDRESS.
020614  000000                                    0                     ;3RD WORD IS THE BUFFER HIGH ADDRESS.
```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```
3631 020616  000000                          0                       ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
3632
3633                                ;       WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
3634
3635 020620  000000                WSMBK:: 0                       ;1ST WORD:: SEL 0
3636 020622  000000                        0                       ;2ND WORD:: SEL 2
3637 020624  000000                        0                       ;3RD WORD:: SEL 4
3638                                        .EVEN
3639                                ;*
3640                                ;       SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
3641                                ;
3642                                ;
3643                                ;INPUTS:
3644                                ;OUTPUTS:
3645                                ;       The NXMFLG is set if we can test.
3646                                ;       The NXMLO and NXMHI addresses are setup.
3647                                ;-
3648
3649 020626                         MEMCK::
3650
3651 020626                                SAVREG                          ;SAVE THE REGISTERS
3652 020632  005037  003136                CLR     NXMFLG                  ;CLEAR THE FLAG
3653 020636  005037  003140                CLR     NXMLO                   ;CLEAR THE TEST ADDRESS LO
3654 020642  005037  003142                CLR     NXMHI                   ;CLEAR THE TEST ADDRESS HI
3655 020646  005737  003146                TST     T23B                    ;IS IT A 11/23B?
3656 020652  001407                       BEQ     1$                      ;NO
3657 020654  023727  002120  007777        CMP     L$HIME,#7777            ; GREATER THAN 128K
3658 020662  103406                       BLO     2$                      ; NO
3659 020664  004737  021002                JSR     PC,NXMTST               ;SETUP THE ADDRESS
3660 020670  000427                       BR      13$                     ;SET THE FLAG AND EXIT
3661 020672  005737  003144        1$:     TST     T23A                    ;IS IT A 11/23A?
3662 020676  001413                       BEQ     4$                      ;NO
3663 020700  023727  002120  005777  2$:     CMP     L$HIME,#5777,           ;GREATER THAN 96K
3664 020706  101023                       BHI     14$                     ;YES,23A/23B WITH 128K MEMORY
3665 020710  023727  002120  003777        CMP     L$HIME,#3777            ;GREATER THAN 64K BUT LESS THAN 92K?
3666 020716  103403                       BLO     4$                      ;NO, CHECK 24K
3667 020720  004737  021002                JSR     PC,NXMTST               ;SETUP THE ADDRESS
3668 020724  000411                       BR      13$                     ;SET THE FLAG AND EXIT
3669 020726  023727  002120  001577  4$:     CMP     L$HIME,#1577            ;GREATER THAN 24K BUT LESS THAN 64K?
3670 020734  103410                       BLO     14$                     ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
3671 020736  004737  021002                JSR     PC,NXMTST               ;SETUP THE ADDRESS
3672 020742  062737  000077  003142        ADD     #77,NXMHI               ;FOOL THE 11/02 & 11/03
3673 020750  005237  003136        13$:    INC     NXMFLG                  ;SET THE FLAG
3674 020754  000411                       BR      15$                     ;EXIT
3675 020756  000410                14$:    BR      15$                     ;NOP FOR PRINTOUT
3676 020760                               PRINTF  #NOMEM                  ;TELL THEM & EXIT ***NO PRINT*****
     020760  012746  005460                MOV     #NOMEM,-(SP)
     020764  012746  000001                MOV     #1,-(SP)
     020770  010600                       MOV     SP,R0
     020772  104417                       TRAP    C$PNTF
     020774  062706  000004                ADD     #4,SP
3677 021000  000207                15$:    RTS     PC                      ;RETURN
3678
3679                                ;*
3680                                ;       SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
3681                                ;
3682                                ;OUTPUTS:NXMLO,NXMHI            ;SETUP WITH NXM ADDRESS
```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```
                                        ;
                                        ;-
                                        
021002  013701  002120          NXMTST: MOV     L$HIME,R1               ;GET TOP OF MEMORY
021006  062701  000200                  ADD     #200,R1                 ;MAKE IT I/O BLOCK OR OTHER NXM
021012  042701  000177                  BIC     #177,R1
021016  010102                          MOV     R1,R2                   ;RESAVE RESULTS
        000006                          .REPT   6
                                        ASL     R1                      ;PUT IN PLACE FOR XFER
                                        .ENDR
021034  010137  003140                  MOV     R1,NXMLO                ;SAVE TEST ADDRESS LOW
        000012                          .REPT   10.
                                        ASR     R2                      ;PUT IN PLACE FOR XFER
                                        .ENDR
021064  042702  177700                  BIC     #177700,R2              ;DON'T WANT ILA!
021070  010237  003142                  MOV     R2,NXMHI                ;SAVE TEST ADDRESS HIGH
021074  000207                          RTS     PC                      ;RETURN

021076                                  ENDMOD
                                        .TITLE  TSV4 - MISCELLANEOUS SECTIONS

021076                                  BGNMOD  TSV4
021076                          TSV4::


                                        .SBTTL  PROTECTION TABLE
021076                                  BGNPROT
021076                          L$PROT::
021076  177777  177777  177777          .WORD   -1, -1, -1,  1          ;NO DEVICE PROTECTION REQUIRED.
021106                                  ENDPROT
```

INITIALIZE SECTION

```
                                        .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
                                ;-
021106                                  BGNINIT
021106                          L$INIT::
021106  005037  002226          40$:    CLR     EXTFEA
021112  005037  003136                  CLR     NXMFLG
021116  012737  006360  002200          MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021124  005037  003154                  CLR     SIFLAG                  ;CLEAR "SOFT INIT" FLAG
021130  005037  003134                  CLR     KTENABLE                ;CLEAR TEST ABOVE 28K FLAG
021134  005037  002302                  CLR     RAMSIZ                  ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021140                                  READEF  #EF.CONTINUE
021140  012700  000036                  MOV     #EF.CONTINUE,R0
021144  104447                          TRAP    C$REFG
021146                                  BNCOMPLETE 1$
021146  103023                          BCC     1$
021150  023737  002202  002012          CMP     UNITN,L$UNIT            ;UNIT IN RANGE?
021156  103070                          BHIS    4$                      ;BR IF NO.
021160  005737  003112                  TST     DUFLG                   ;DROPPED UNIT?
021164  100472                          BMI     NXTU                    ;BR IF YES
021166  013701  002202                  MOV     UNITN,R1
021172  006301                          ASL     R1
021174  005761  003176                  TST     ERTABL(R1)
021200  001516                          BEQ     SETU
021202  032761  040000  0031^6          BIT     #BIT14,ER^ABL(R1)       ;DROPPED?
021210  001060                          BNE     NXTU
021212                                  EXIT    INIT                    ;DO NOTHING IF "CONTINUE".
021212  104432                          TRAP    C$EXIT
021214  000416                          .WORD   L10030-.
021216                          1$:     READEF  #EF.NEW
021216  012700  000035                  MOV     #EF.NEW,R0
021222  104447                          TRAP    C$REFG
021224                                  BNCOMPLETE NXTU                 ;TAKE NEXT UNIT IF NOT NEW PASS.
021224  103052                          BCC     NXTU
021226                                  READEF  #EF.START
021226  012700  000040                  MOV     #EF.START,R0
021232  104447                          TRAP    C$REFG
021234                                  BCOMPLETE 2$
021234  103404                          BCS     2$
021236                                  READEF  #EF.RESTART
021236  012700  000037                  MOV     #EF.RESTART,R0
021242  104447                          TRAP    C$REFG
021244                                  BNCOMPLETE 31$
021244  103031                          BCC     31$
021246                          2$:                                     ;1ST PASS, BUS-INIT...
021246                                  BRESET                          ;BUS RESET.
021246  104433                          TRAP    C$RESET
```

INITIALIZE SECTION

```
3768 021250  005037  002214                    CLR     TSTCNT                  ;NUMBER OF TESTS RUN IN PASS
3769 021254  005037  002222                    CLR     FATFLG                  ;CLEAR FATAL ERROR COUNT
3770 021260  005037  003144                    CLR     T23A                    ;CLEAR 11/23A FLAG
3771 021264  005037  003146                    CLR     T23B                    ;CLEAR 11/23B FLAG
3772                                       ;    MOV     #340,-(SP)
3773                                       ;    MOV     #20$,-(SP)              ;RETURN TO DEBUGGER
3774                                       ;    JMP     0.ODT           ;;@@ENTER THE DEBUGGER
3775 021270  005037  003400                    CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
3776 021274                             20$:
3777 021274  012737  177777  002204              MOV     #-1,@VP                 ;...QUICK VERIFY...
3778 021302  004737  020112                    JSR     PC,ENVIRN               ;SET ENVIRONMENT.
3779 021306  004737  020272                    JSR     PC,KTINIT               ;INITIALIZE KT MEMORY MANAGEMENT
3780 021312  012700  003176                    MOV     #ERTABL,R0
3781 021316  005020                     30$:   CLR     (R0)+                   ;CLEAR THE ERROR TABLE
3782 021320  020027  003376                    CMP     R0,#ERTABE
3783 021324  103774                          BLO     30$
3784 021326  000404                          BR      4$
3785 021330  005037  002204              31$:   CLR     @VP
3786 021334  000137  021404                    JMP     PASRPT                  ;GO REPORT THE STATUS
3787
3788 021340                             4$:
3789 021340  012737  177777  002202    NEWPAS: MOV     #-1,UNITN               ;INIT UNIT NUMBER...
3790 021346  005037  002220                    CLR     DEVCNT                  ;CLEAR COUNT OF DEVICES RUNNING
3791 021352                             NXTU:  BREAK
     021352  104422                          TRAP    C$BRK
3792 021354  005237  002202                    INC     UNITN                   ;...AND SET NEXT UNIT NUMBER.
3793 021360  023737  002202  002012            CMP     UNITN,L$UNIT
3794 021366  103423                          BLO     SETU
3795 021370  012737  177777  003112            MOV     #-1,DUFLG
3796 021376  000401                          BR      11$
3797 021400                                   DOCLN                           ;ABORT, NO MORE UNITS.
     021400  104444                          TRAP    C$DCLN
3798 021402  000240                     11$:   NOP
3799 021404                             PASRPT:
3800 021404  023727  002012  000001            CMP     L$UNIT,#1               ;HOW MANY UNITS SELECTED?
3801 021412  101752                          BLOS    NEWPAS                  ;BR IF ONLY 1
3802 021414  005737  002220                    TST     DEVCNT                  ;ARE ANY STILL RUNNING?
3803 021420  001747                          BEQ     NEWPAS                  ;BR IF NO
3804 021422                                   RFLAGS  R0
     021422  104421                          TRAP    C$RFLA
3805 021424  032700  000100                    BIT     #ISR,R0                 ;SHOULD WE PRINT STATISTICS
3806 021430  001343                          BNE     NEWPAS                  ;BR IF NO
3807
3808 021432                                   DORPT
     021432  104424                          TRAP    C$DRPT
3809 021434  000741                          BR      NEWPAS
3810 021436                             10$:
3811
3812 021436                             SETU:  GPHARD  UNITN,R0                ;GET UNIT N P-TABLE POINTER.
     021436  013700  002202                    MOV     UNITN,R0
     021442  104442                          TRAP    C$GPHRD
3813 021444                                   BNCOMPLETE NXTU                 ;BR IF UNIT NOT AVAILABLE.
     021444  103342                          BCC     NXTU
3814 021446  005037  003112                    CLR     DUFLG                   ;CLEAR "DROPPED" FLAG.
3815 021452  005237  002220                    INC     DEVCNT
3816 021456  012001                          MOV     (R0)+,R1                ;GET 1ST REGISTER ADDRESS.
3817 021460  010137  002206                    MOV     R1,CSRADDR              ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
```

INITIALIZE SECTION

```
                                        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     IPRI,(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.
021532  104421                          TRAP    C$RFLA
                                        BIT     #PNT,R0                 ;SHOULD WE PRINT UNIT #?
                                        BEQ     10$                     ;BR IF NOT.
                                        PRINTF  #PUNIT,UNITN            ;PRINT THE UNIT #
021542  013746  002202                  MOV     UNITN,-(SP)
021546  012746  021634                  MOV     #PUNIT,-(SP)
021552  012746  000002                  MOV     #2,-(SP)
021556  010600                          MOV     SP,R0
021560  104417                          TRAP    C$PNTF
021562  062706  000006                  ADD     #6,SP
021566                          10$:
021566  005037  003114                  CLR     NODEV
021572  013701  002206                  MOV     CSRADDR,R1              ;ADDRESS OF FIRST REGISTER
021576  010102                          MOV     R1,R2                   ;START OF REGISTERS
021600  062702  000002                  ADD     #TSSR,R2                ;ADDRESS OF TSSR REGISTER
021604  004737  016376                  JSR     PC,XNXM                 ;TEST BOTH CONTROLLER REGISTERS...
021610  103005                          BCC     2$                      ;...AND BR IF ALL OK.
021612  010137  003114                  MOV     R1,NODEV                ;FLAG DEVICE AS NON-EXISTENT
021616  012737  177777  003112          MOV     #-1,DUFLG               ;DROP THIS UNIT.
021624                          2$:
                                ;
                                ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
                                ;
021624                          5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
021624  012700  000000                  MO      #PRI00,R0
021630  104441                          TRAP    C$SPRI
021632                                  ENDINIT
021632                          L10030:
021632  104411                          TRAP    C$INIT

021634      045     116     045 PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
                                        .EVEN
```

```
                                        .SBTTL  ADD AND DROP UNITS SECTIONS

                                ;++
                                ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
                                ;    OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
                                ;--
021702                                  BGNAU
021702                          L$AU::
021702  010001                        MOV     R0,R1                   ; GET UNIT TO BE ADDED (R0)
021704  006301                        ASL     R1                      ; MAKE IT A WORD INDEX
021706  052761  100000  003176        BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
021714  042761  040000  003176        BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
021722                              PRINTF  #1$,R0
021722  010046                      MOV     R0,-(SP)
021724  012746  021750              MOV     #1$,-(SP)
021730  012746  000002              MOV     #2,-(SP)
021734  010600                      MOV     SP,R0
021736  104417                      TRAP    C$PNTF
021740  062706  000006              ADD     #6,SP
021744                              EXIT    AU
021744  000167                      .WORD   J$JMP
021746  000026                      .WORD   L10031-2-.
021750     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN

                                        ENDAU           ; UNUSED.
021776                          L10031:
021776  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.

022000                                  BGNDU
022000                          L$DU::
022000  012737  177777  003112        MOV     #-1,DUFLG
022006  010001                      MOV     R0,R1
022010  006301                      ASL     R1
022012  052761  140000  003176        BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022020  000240  000240  000240        240,240,240                     ; ?????????
022026                              PRINTF  #1$,R0
022026  010046                      MOV     R0,-(SP)
022030  012746  022054              MOV     #1$,-(SP)
022034  012746  000002              MOV     #2,-(SP)
022040  010600                      MOV     SP,R0
022042  104417                      TRAP    C$PNTF
022044  062706  000006              ADD     #6,SP
022050                              EXIT    DU
022050  000167                      .WORD   J$JMP
022052  000030                      .WORD   L10032-2-.
```

ADD AND DROP UNITS SECTIONS

```
   3900 022054     045     116     045  1$:     .ASCIZ  /<N><A>  UNIT <D><A> DROPPED/
   3901                                         .EVEN
   3902 022104                                  ENDDU
        022104                          L10032:
        022104  104453                         TRAP    C$DU
   3903                                 ;++
   3904                                 ; AUTO-DROP CODE SECTION.
   3905                                 ;--
   3906 022106                                  BGNAUTO
        022106                          L$AUTO::
   3907 022106  013705  002206                 MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
   3908 022112  012703  000550                 MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
   3909 022116  004737  016250          10$:   JSR     PC,WAITF                ;WAIT FOR SSR TO SET
   3910 022122  103420                         BCS     20$                     ;LEAVE WHEN SSR IS SET
   3911 022124                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
        022124  012727  000372                 MOV     #250.,(PC)+
        022130  000000                         .WORD   0
        022132  013727  002116                 MOV     L$DLY,(PC)+
        022136  000000                         .WORD   0
        022140  005367  177772                 DEC     -6(PC)
        022144  001375                         BNE     .-4
        022146  005367  177756                 DEC     -22(PC)
        022152  001367                         BNE     .-20
   3912 022154  005303                         DEC     R3                      ;BUMP COUNTER DOWN
   3913 022156  001357                         BNE     10$                     ;KEEP GOING
   3914 022160  004737  017202                 JSR     PC,CKDROP               ;TRY AND DROP UNIT
   3915 022164                          20$:
   3916 022164                                  ENDAUTO                 ; UNUSED.
        022164                          L10033:
        022164  104461                         TRAP    C$AUTO
```

CLEAN-UP AND REPORT CODING SECTIONS

```
                                        .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.
                                ;--
                                        BGNCLN
                                L$CLEAN::
                                        MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
                                        TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
                                        BMI     1$              ;...AND GROSS CONTROLLER FAULT...
                                                                ;...DON'T TRY TO XCT CLEANUP CODE.

                                        MOV     #0,TSSR(R5)     ;DO SOFT INIT
                                        JSR     PC,WAITF
                                1$:
                                2$:     ENDCLN
                                L10034:
                                        TRAP    C$CLEAN

                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--
                                        BGNRPT
                                L$RPT::
                                        PRINTS  #DEVSUM
                                        MOV     #DEVSUM,-(SP)
                                        MOV     #1,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTS
                                        ADD     #4,SP
                                        MOV     R2,-(SP)
                                        MOV     R3,-(SP)
                                        MOV     R4,-(SP)
                                        MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
                                        CLR     R3              ; CLEAR UNIT NUMBER
                                1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
                                        BEQ     4$              ; ZERO IF UNIT NOT RUN
                                        BPL     4$
                                        BIT     #BIT14,R2       ; WAS UNIT DROPPED?
                                        BNE     2$              ; BR IF YES
                                        BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
                                        PRINTS  #DEVONL,R3,R2   ; PRINT
                                        MOV     R2,-(SP)
                                        MOV     R3,-(SP)
                                        MOV     #DEVONL,-(SP)
                                        MOV     #3,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTS
                                        ADD     #10,SP
                                        BR      4$
                                2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
                                        BNE     3$              ; BR IF NO
                                        PRINTS  #DEVNXR,R3
                                        MOV     R3,-(SP)
                                        MOV     #DEVNXR,-(SP)
```

CLEAN-UP AND REPORT CODING SECTIONS

```
        022332  012746  000002                  MOV     #2,-(SP)
        022336  010600                           MOV     SP,R0
        022340  104416                           TRAP    C$PNTS
        022342  062706  000006                  ADD     #6,SP
  3957  022346  000431                           BR      4$
  3958  022350  020227  160001          3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
  3959  022354  001012                           BNE     30$             ; BR IF NO.
  3960  022356                                   PRINTS  #DEVNRD,R3
        022356  010346                           MOV     R3,-(SP)
        022360  012746  022645                  MOV     #DEVNRD,-(SP)
        022364  012746  000002                  MOV     #2,-(SP)
        022370  010600                           MOV     SP,R0
        022372  104416                           TRAP    C$PNTS
        022374  062706  000006                  ADD     #6,SP
  3961  022400  000414                           BR      4$
  3962  022402  042702  170000          30$:    BIC     #^C7777,R2
  3963  022406                                   PRINTS  #DEVDRO,R3,R2
        022406  010246                           MOV     R2,-(SP)
        022410  010346                           MOV     R3,-(SP)
        022412  012746  022726                  MOV     #DEVDRO,-(SP)
        022416  012746  000003                  MOV     #3,-(SP)
        022422  010600                           MOV     SP,R0
        022424  104416                           TRAP    C$PNTS
        022426  062706  000010                  ADD     #10,SP
  3964  022432  062704  000002          4$:     ADD     #2,R4
  3965  022436  005203                           INC     R3
  3966  022440  020427  003376                  CMP     R4,#ERTABE
  3967  022444  103701                           BLO     1$
  3968  022446  012604                           MOV     (SP)+,R4
  3969  022450  012603                           MOV     (SP)+,R3
  3970  022452  012602                           MOV     (SP)+,R2
  3971  022454                                   ENDRPT          ; UNUSED.
        022454                          L10035:
        022454  104425                           TRAP    C$RPT
  3972
  3973  022456     045     116     045  DEVSUM: .ASCIZ  /<N><A>DEVICE STATUS SUMMARY:<N>/
  3974  022513     045     101     040  DEVONL: .ASCIZ  /<A>  UNIT <D3><A>   ONLINE,  ERRORS = <D><N>/
  3975  022563     045     101     040  DEVNXR: .ASCIZ  /<A>  UNIT <D3><A>   DROPPED, NON-EXISTENT REGISTER<N>/
  3976  022645     045     101     040  DEVNRD: .ASCIZ  /<A>  UNIT <D3><A>   DROPPED, NOT READY AT STARTUP<N>/
  3977  022726     045     101     040  DEVDRO: .ASCIZ  /<A>  UNIT <D3><A>   DROPPED, ERRORS = <D><N>/
  3978                                           .EVEN
  3979
  3980  022776                                   ENDMOD
  3981
```

CLEAN-UP AND REPORT CODING SECTIONS

```
                                                .TITLE  TSV5 - HARDWARE TESTS
                                                BGNMOD  TSV5
022776                                  TSV5::
```

```
                                        .SBTTL  TEST  1: INITIALIZE AFTER WRITE CHARACTERISTICS
                                ;+
                                ; TEST DESCRIPTION:
                                ;
                                ;       This test verifies that a Hardware Initialize command
                                ;       invoked after a Write Characteristics command sets up
                                ;       the Command, Message and Characteristic image blocks
                                ;       in the controller ram correctly.
                                ;
                                ; TEST STEPS:
                                ;
                                ;       REPEAT FOR LOOPCNT
                                ;         BEGIN
                                ;               Do WRITE CHARACTERISTICS command.
                                ;               If the NBA bit in the TSSR register is NOT=0 then Print Error.
                                ;               Write to TSSR register to soft initialize the controller
                                ;               If controller RAM 310-377 NOT=0 then Print Error
                                ;         END
                                ;--

022776                                  BGNTST
022776                                                                          T1::
022776  012700  023432                   MOV     #TST13ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
023002  004737  016510                   JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
023006  012737  000012  002216           MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
023014                          T13LOOP:
023014  004737  023706                   JSR     PC,T13REST              ;SET PACKET TO START-UP VALUES

023020  012703  002764                   MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
023024  012704  023370                   MOV     #T13PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
023030  012764  000010  000006           MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
023036                          5$:
023036  004737  015774                   JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
023042  103405                           BCS     10$                     ;BR IF SOFT INIT OKAY
023044  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023046                                   ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
023046  104455                                                                   TRAP    C$ERDF
023050  000144                                                                   .WORD   100
023052  003652                                                                   .WORD   SFIERR
023054  012034                                                                   .WORD   SFIMSG

                                ;Do WRITE CHARACTERISTICS command.
023056  005037  002222           10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
023062  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
023066  004737  016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
023072                                   FORCERROR       12$                     ;BBDFORCE ERROR IF FORCER=1
023106  103407                           BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
023110  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023112                                   NEXT.ERRNO
023112                          12$:     ERRDF   ERRNO,T13SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
023112  104455                                                                   TRAP    C$ERDF
023114  000145                                                                   .WORD   101
023116  023617                                                                   .WORD   T13SSR
023120  012046                                                                   .WORD   PKTSSR
023122  005237  002222                   INC     FATFLG                  ;SET FATAL ERROR FLAG
023126                          15$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```
      023126  104406                                                                  TRAP    C$CLP1
 4063 023130  016501  000002                MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 4064 023134  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 4065 023140  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 4066 023144  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 4067 023146  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 4068
 4069                              ;If the NBA bit in the TSSR register is NOT=0 then Print Error.
 4070 023152                       25$:
 4071 023152                              FORCERROR       27$                     ;@@D
 4072 023166  020201                      CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 4073 023170  001404                      BEQ     30$                     ;OKAY IF MATCH
 4074 023172                              NEXT.ERRNO
 4075 023172                       27$:   ERRHRD  ERRNO,T13NBA,PKTSSR     ;NBA NOT ZERO
      023172  104456                                                                  TRAP    C$ERHRD
      023174  000146                                                                  .WORD   102
      023176  023544                                                                  .WORD   T13NBA
      023200  012046                                                                  .WORD   PKTSSR
 4076 023202                       30$:   CKLOOP                          ;LOOP ON ERROR ?
      023202  104406                                                                  TRAP    C$CLP1
 4077
 4078                              ;Write to TSSR register to soft initialize the controller
 4079 023204                       40$:
 4080 023204  004737  015774               JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
 4081 023210                              FORCERROR       42$                     ;@@D
 4082 023224  103405                      BCS     50$                     ;BR IF SOFT INIT OKAY
 4083 023226  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4084 023230                              NEXT.ERRNO
 4085 023230                       42$:   ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      023230  104455                                                                  TRAP    C$ERDF
      023232  000147                                                                  .WORD   103
      023234  003652                                                                  .WORD   SFIERR
      023236  012034                                                                  .WORD   SFIMSG
 4086
 4087                              ;If controller RAM 310-377 NOT=0 then Print Error
 4088 023240  012704  000310        50$:   MOV     #310,R4                 ;START WITH LOC 310
 4089 023244  005002                       CLR     R2                      ;MEMORY EXPECTED SHOULD BE 000000
 4090 023246  105065  000000               CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
 4091 023252  004737  016336               JSR     PC,CHKTSSR              ;WAIT FOR SSR READY
 4092 023256  010465  000000        60$:   MOV     R4,TSDB(R5)             ;SELECT RAM ADDRESS
 4093 023262  004737  016336               JSR     PC,CHKTSSR              ;WAIT FOR SSR READY
 4094 023266  116501  000000               MOVB    TSBA(R5),R1             ;READ LOC CONTENTS
 4095 023272                              FORCERROR       62$,NOTSSR              ;@@D
 4096 023302  120102                      CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
 4097 023304  001406                      BEQ     70$                     ;BRANCH IF DATA OKAY
 4098 023306                              NEXT.ERRNO
 4099 023306                       62$:   ERRDF   ERRNO,T13MEM,RAMEXP     ;MEMORY NOT ZERO AFTER INIT.
      023306  104455                                                                  TRAP    C$ERDF
      023310  000150                                                                  .WORD   104
      023312  023505                                                                  .WORD   T13MEM
      023314  015530                                                                  .WORD   RAMEXP
 4100 023316  005237  002222               INC     FATFLG                  ;SET THE FATAL ERROR FLAG
 4101 023322                       70$:   CKLOOP
      023322  104406                                                                  TRAP    C$CLP1
 4102 023324                              ESCAPE  TST                     ;EXIT ON FATAL ERROR
      023324  104410                                                                  TRAP    C$ESCAPE
      023326  000426                                                                  .WORD   L10036-.
```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```
4103
4104 023330  005204                  82$:    INC     R4                      ;LOOK AT NEXT RAM LOC.
4105 023332  020427  000400                  CMP     R4,#400                 ;AT TOP OF RAM ADDRESS SPACE
4106 023336  001347                          BNE     60$                     ;BRANCH TILL ALL MEMORY TESTED
4107
4108
4109 023340  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
4110 023344  001402                          BEQ     160$                    ;BRANCH IF NOT
4111 023346  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4112 023352  004737  016456          160$:   JSR     PC,TSTLOOP              ;DONE ALL ITERATIONS?
4113 023356  103002                          BCC     165$                    ;BR IF YES
4114 023360  000137  023014                  JMP     T13LOOP                 ;LOOP UNTIL ITERATION COUNT DONE
4115 023364                          165$:
4116 023364                                  EXIT    TST
     023364  104432                                                          TRAP    C$EXIT
     023366  000366                                                          .WORD   L10036-.
4117
4118
4119                                 ;+
4120                                 ;LOCAL STORAGE FOR THIS TEST
4121                                 ;-
4122
4126 023370                          T13PACKET:                              ;COMMAND PACKET FOR TEST
4127 023370  100004                          .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
4128 023372  023400                          .WORD   T13DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
4129 023374  000000                          .WORD   0
4130 023376  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
4131
4132 023400                          T13DATA:                                        ;CHARACTERISTICS DATA BLOCK
4133 023400  023412                          .WORD   T13BFR                  ;ADDRESS OF MESSAGE BUFFER
4134 023402  000000                          .WORD   0
4135 023404  000016                          .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
4136 023406  000000  000000                  .WORD   0,0
4137
4138 023412                          T13BFR: .BLKW   8.                      ;MESSAGE BUFFER
4139                                 ;LOCAL TEXT MESSAGES FOR TEST
4140                                 ;-
4141
4142 023432     111     156     151  TST13ID:        .ASCIZ  'Initialization After WRITE CHARACTERISTICS'
4143 023505     111     156     143  T13MEM: .ASCIZ  'Incorrect RAM Data After Init'
4144                                         .EVEN
4145 023544     127     122     111  T13NBA: .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
4146 023617     103     157     156  T13SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
4147
4148
4149                                 ;+
4150                                 ;
4151                                 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4152                                 ;
4153                                 ;-
4154
4155                                 .EVEN
4156
4157 023706                          T13REST:
4158 023706                                  SAVREG                          ;SAVE THE REGISTERS
4159 023712  012701  023370                  MOV     @T13PACKET,R1           ;START OF THE PACKET
4160 023716  012721  100004                  MOV     @100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```
 4161 023722  012721  023400                  MOV     #T13DATA,(R1)+  ;ADDRESS OF CHAR DATA BLOCK
 4162 023726  005021                          CLR     (R1)+           ;EXTENDED ADDRESS
 4163 023730  012721  000010                  MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
 4164 023734  012721  023412                  MOV     #T13BFR,(R1)+   ;ADDRESS OF MESSAGE BUFFER
 4165 023740  005021                          CLR     (R1)+
 4166 023742  012721  000016                  MOV     #14.,(R1)+      ;LENGTH OF MESSAGE BUFFER
 4167 023746  005021                          CLR     (R1)+
 4168 023750  005011                          CLR     (R1)
 4169 023752  000207                          RTS     PC              ;RETURN
 4170 023754                                  ENDTST
      023754                                                                  L10036:
      023754  104401                                                                TRAP    C$ETST
 4171
 4172                                         .SBTTL  TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND
 4173                                ;+
 4174                                ;
 4175                                ;THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A
 4176                                ;BSELO SELECT CODE OF 0 (NO-OP) EXECUTES CORRECTLY. IT ALSO
 4177                                ;VERIFIES THAT A WRITE SUBSYSTEM MEMORY COMMAND WITH A NON-ZERO
 4178                                ;MODE FIELD IS REJECTED. THE TEST FURTHER VERIFIES MICROPROGRAM
 4179                                ;COMMAND DECODING AND HANDLING SEQUENCES.
 4180                                ;
 4181                                ;-
 4182 023756                                  BGNTST
      023756                                                                  T2::
 4187 023756  012700  025571                  MOV     #TST14ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
 4188 023762  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
 4189 023766  012737  000024  002216          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
 4190 023774                         T14LOOP:
 4191 023774                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      023774                                                                  T2.1:
      023774  104402                                                                TRAP    C$BSUB
 4192 023776  004737  025636                  JSR     PC,T14REST              ;SET PACKET TO INITIAL VALUES
 4193 024002  004737  025674                  JSR     PC,T14RST               ;SET PACKET TO INITIAL VALUES
 4194 024006                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      024006  012700  000000                                                          MOV     #PRI00,R0
      024012  104441                                                                  TRAP    C$SPRI
 4195 024014                         5$:
 4196 024014                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      024014  104404                                                                 TRAP    C$BSEG
 4197 024016  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 4198 024022  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
 4202 024024  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4203 024026                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      024026  104455                                                                  TRAP    C$ERDF
      024030  000311                                                                  .WORD   201
      024032  003652                                                                  .WORD   SFIERR
      024034  012034                                                                  .WORD   SFIMSG
 4204 024036                         10$:
 4205 024036  012704  025130                  MOV     #T14PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 4206 024042  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 4207 024046  103405                          BCS     11$                     ;BR, IF COMMAND ISSUED OK
 4211 024050  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4212 024052                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      024052  104456                                                                  TRAP    C$ERHRD
      024054  000312                                                                  .WORD   202
      024056  005056                                                                  .WORD   WRTMSG
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
        024060  012034                                                          .WORD   SFIMSG
4213    024062                  11$:
4214    024062  005037  002222          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
4215    024066  005037  002224          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
4216    024072  012704  024510          MOV     #T14PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
4217    024076  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
4218    024102  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4219    024106  103407                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
4220    024110  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4224    024112                          ERRDF   ERRNO,T14SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        024112  104455                                                          TRAP    C$ERDF
        024114  000313                                                          .WORD   203
        024116  025323                                                          .WORD   T14SSR
        024120  012046                                                          .WORD   PKTSSR
4225    024122  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
4226    024126                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        024126  104406                                                          TRAP    C$CLP1
4227    024130                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
        024130  104410                                                          TRAP    C$ESCAPE
        024132  000074                                                          .WORD   10000$-.
4228    024134  005737  002224          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
4229    024140  001004                  BNE     22$                     ;BRANCH IF YES
4233    024142                          ERRHRD  ERRNO,T14NINT,PKTSSR
        024142  104456                                                          TRAP    C$ERHRD
        024144  000314                                                          .WORD   204
        024146  025413                                                          .WORD   T14NINT
        024150  012046                                                          .WORD   PKTSSR
4234    024152  016501  000002  22$:    MOV     SSR(R5),R1              ;GET THE CONTENTS OF TSSR
4235    024156  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
4236    024162  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
4237    024166  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
4238    024170  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
4239    024174  020201          25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
4240    024176  001404                  BEQ     30$                     ;OKAY IF MATCH
4244    024200                          ERRHRD  ERRNO,T14NBA,PKTSSR     ;NBA NOT ZERO
        024200  104456                                                          TRAP    C$ERHRD
        024202  000315                                                          .WORD   205
        024204  025152                                                          .WORD   T14NBA
        024206  012046                                                          .WORD   PKTSSR
4245    024210                  30$:
4246    024210  004737  011114  35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
4247    024214  103405                  BCS     59$                     ;RAM OK GO ON
4251    024216                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
        024216  104456                                                          TRAP    C$ERHRD
        024220  000316                                                          .WORD   206
        024222  004745                                                          .WORD   PKTRAM
        024224  015510                                                          .WORD   RAMERR
4252    024226                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        024226                                                                  10000$:
        024226  104405                                                          TRAP    C$ESEG
4253
4254
4255    024230                  59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        024230                                                                  L10040:
        024230  104403                                                          TRAP    C$ESUB
4256
4257    024232  005737  002222          TST     FATFLG                  ;ANY FATAL ERRORS ?
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
4258 024236  001402                    BEQ     60$                     ;BRANCH IF NOT
4259 024240  004737  017202            JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4260 024244                        60$:
4261                                   ;*
4262                                   ;
4263                                   ;TEST 2, SUBTEST 2
4264                                   ;
4265                                   ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
4266                                   ;WRITE SUBSYSTEM MEMORY COMMAND TO BE REJECTED
4267                                   ;
4268                                   ;-
4269
4270 024244                            BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
     024244                                                                    T2.2:
     024244  104402                                                                 TRAP    C$BSUB
4271
4272 024246                            SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     024246  012700  000000                                                         MOV     #PRI00,R0
     024252  104441                                                                 TRAP    C$SPRI
4273 024254  012704  025130        5$: MOV     #T14PK2,R4              ;GET THE ADDRESS OF COMMAND PACKET
4274 024260  004737  025636            JSR     PC,T14REST              ;RESTORE PACKET TO STARTING VALUES
4275 024264  004737  025674            JSR     PC,T14RST               ;RESTORE PACKET TO STARTING VALUES
4276 024270                            BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     024270  104404                                                                 TRAP    C$BSEG
4277
4278 024272  004737  015774            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
4279 024276  103405                    BCS     10$                     ;BR IF SOFT INIT = OK
4283 024300  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4284 024302                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     024302  104455                                                                 TRAP    C$ERDF
     024304  000317                                                                 .WORD   207
     024306  003652                                                                 .WORD   SFIERR
     024310  012034                                                                 .WORD   SFIMSG
4285 024312                        10$:
4286 024312  004737  010662            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
4287 024316  103405                    BCS     11$                     ;BR, IF COMMAND ISSUED OK
4291 024320  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4292 024322                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE SUBSYSTEM MEMORY FAILED
     024322  104456                                                                 TRAP    C$ERHRD
     024324  000320                                                                 .WORD   208
     024326  005056                                                                 .WORD   WRTMSG
     024330  012034                                                                 .WORD   SFIMSG
4293 024332                        11$:
4294 024332  005037  002224            CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
4295 024336  012704  024510            MOV     #T14PACKET,R4           ;SET-UP WITH WRT SUBSYS MEM PACKET
4296 024342  052714  007000            BIS     #007000,(R4)            ;NON-ZERO COMMAND MODE BITS
4297 024346  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
4298 024352  004737  016250            JSR     PC,WAITF                ;WAIT FOR SSR TO SET
4299 024356  103405                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
4300 024360  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4304 024362                            ERRDF   ERRNO,T14SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     024362  104455                                                                 TRAP    C$ERDF
     024364  000321                                                                 .WORD   209
     024366  025323                                                                 .WORD   T14SSR
     024370  012046                                                                 .WORD   PKTSSR
4305 024372                        15$:CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     024372  104406                                                                 TRAP    C$CLP1
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
4306 024374                                ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
     024374  104410                                                                TRAP    C$ESCAPE
     024376  000074                                                                .WORD   10000$-.
4307 024400  005737  002224                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
4308 024404  001004                        BNE     22$                     ;BRANCH IF YES
4312 024406                                ERRHRD  ERRNO,T14NINT,PKTSSR
     024406  104456                                                                TRAP    C$ERHRD
     024410  000322                                                                .WORD   210
     024412  025413                                                                .WORD   T14NINT
     024414  012046                                                                .WORD   PKTSSR
4313 024416  016501  000002          22$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
4314 024422  012702  100206                MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
4315 024426  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
4316 024432  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
4317 024434  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
4318 024440  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
4319 024442  001404                        BEQ     30$                     ;OKAY IF MATCH
4323 024444                                ERRHRD  ERRNO,T142REJ,PKTSSR    ;COMMAND NOT REJECTED
     024444  104456                                                                TRAP    C$ERHRD
     024446  000323                                                                .WORD   211
     024450  025226                                                                .WORD   T142REJ
     024452  012046                                                                .WORD   PKTSSR
4324 024454                          30$:
4325 024454  004737  011114          35$:  JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
4326 024460  103405                        BCS     59$                     ;RAM OK GO ON
4330 024462                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     024462  104456                                                                TRAP    C$ERHRD
     024464  000324                                                                .WORD   212
     024466  004745                                                                .WORD   PKTRAM
     024470  015510                                                                .WORD   RAMERR
4331 024472                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     024472                                                                 10000$:
     024472  104405                                                                TRAP    C$ESEG
4332
4333 024474                          59$:  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     024474                                                                 L10041:
     024474  104403                                                                TRAP    C$ESUB
4334 024476                                EXIT    TST                     ;ALL DONE THIS TEST
     024476  104432                                                                TRAP    C$EXIT
     024500  001246                                                                .WORD   L10037-.
4335
4336                                       ;+
4337                                       ;LOCAL STORAGE FOR THIS TEST
4338                                       ;-
4339
4341         024510                                .=<.+10>&177770
4343 024510                          T14PACKET:                            ;COMMAND PACKET FOR TEST
4344 024510  100206                        .WORD   100206                  ;WRITE SUBSYSTEM MEM. COMMAND, WITH IE, ACK
4345 024512  024520                        .WORD   T14DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
4346 024514  000000                        .WORD   0
4347 024516  000006                        .WORD   6.                      ;STARTING VALUE OF BLOCK SIZE
4348 024520                          T14DATA:                              ;CHARACTERISTICS DATA BLOCK
4349 024520     000                  T14BS0: .BYTE   0                       ;BSEL0 BYTE
4350 024521     000                  T14BS1: .BYTE   0                       ;BSEL1 BYTE
4351 024522  000000                T14BS2: .WORD   0                       ;BSEL1 WORD
4352 024524  000000                        .WORD   0                       ;DATA
4353 024526                          T14BFR: .BLKW   128.                  ;MESSAGE BUFFER
```

```
                                                .=<.+10>&177770
                                        T14PK2:                         ;COMMAND PACKET F R TEST
                                                .WORD   100204          ;WRITE CHARA. MEM. CMND., WITH IE, ACK
                                                .WORD   T14DTA          ;ADDRESS OF SELECT DATA BLOCK
                                                .WORD   0
                                                .WORD   8.              ;STARTING VALUE OF BLOCK SIZE

                                        T14DTA:                         ;SELECT DATA BLOCK
                                                .WORD   T14BFR          ;ADDRESS OF MESSAGE BUFFER
                                                .WORD   0
                                                .WORD   256.            ;LENGTH OF MESSAGE BUFFER
                                                .WORD   0,0

                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

                127     122     111     T14NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
                127     122     111     T142REJ:        .ASCIZ  'WRITE SUBSYSTEM MEMORY Not Rejected With Non-Zero Mode Field'
                103     157     156     T14SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
                105     170     160     T14NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
                111     156     143     T14TSBA:        .ASCIZ  'Incorrect TSBA Address After WRITE SUBSYSTEM MEMORY'
                102     141     163     TST14ID:        .ASCIZ  'Basic WRITE SUBSYSTEM MEMORY Command'
                                                .EVEN


                                        ;+
                                        ;
                                        ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
                                        ;WRITE SUBSYSTEM MEMORY COMMAND
                                        ;
                                        ;-

                                        T14REST:
                                                SAVREG                  ;SAVE THE REGISTERS
                                                MOV     #T14PACKET,R1   ;START OF THE PACKET
                                                MOV     #100206,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK, IE
                                                MOV     #T14DATA,(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

                                        T14RST:
                                                SAVREG                  ;SAVE THE REGISTERS
                                                MOV     #T14PK2,R1      ;START OF THE PACKET
                                                MOV     #100204,(R1)+   ;WRITE CHARA. WITH ACK, IE
                                                MOV     #T14DTA,(R1)+   ;ADDRESS OF CHARAISTICS DATA BLOCK
                                                CLR     (R1)+           ;EXTENDED ADDRESS
                                                MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
```

SEQ 0110

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
4413 025722  012721  024526                MOV     #T14BFR,(R1)+       ;MESSAGE BUFFER ADDRESS
4414 025726  005021                        CLR     (R1)+
4415 025730  012721  000400                MOV     #256.,(R1)+         ;LENGTH OF MESSAGE BUFFER
4416 025734  005021                        CLR     (R1)+
4417 025736  005011                        CLR     (R1)
4418 025740  005037  024526                CLR     T14BFR              ;CLEAR 1ST LOC IN MESSAGE BUFFER
4419 025744  000207                        RTS     PC                  ;RETURN
4420 025746                                ENDTST
     025746
     025746  104401                                        L10037:  TRAP    C$ETST
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
                                        .SBTTL  TEST  3: DMA MEMORY ADDRESSING

                                ;++
                                ; TEST 3
                                ;
                                ; TEST DESCRIPTION
                                ;
                                ;       This test verifies that the controller can properly address and
                                ;       access all available CPU memory (other than that occupied by the
                                ;       diagnostic and diagnostic supervisor code) for both reading (DATI)
                                ;       and writing (DATO). Verified are the LSI-11 Bus drivers for all
                                ;       available address lines. Up to this point only 16 bits have been
                                ;       used for DMA transfers.
                                ;
                                ; TEST STEPS
                                ;
                                ;  REPEAT FROM 1 TO LOOPCNT
                                ;    BEGIN
                                ;       Do Subtest 1 - Verify GET STATUS selected locations
                                ;       Do Subtest 2 - Verify message packets selected locations
                                ;       Do Subtest 3 - Verify Characteristic data selected locations
                                ;       Do Subtest 4 - Verify NXM to selected invalid addresses
                                ;    END
                                ;
                                ;--

                                        BGNTST
025750                                                                          T3::
025750
025750  012700  030020                  MOV     @TST12ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
025754  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
025760  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
025766  005237  003150                  INC     T3BFLG                  ;SET TEST FLAG
025772  004737  020626                  JSR     PC,MEMCK                ;CHECK MEMORY

025776                          T12LOOP:                                ;LOOP ON TEST LABEL

                                        .SBTTL  TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS
                                ;++
                                ; TEST 3: SUBTEST 1:
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies the controller can fetch a get status
                                ;       command from all available memory locations.
                                ;       Two word blocks are tested one at a time by first setting
                                ;       all available memory to a background pattern of 125252.
                                ;       A Get Status command is then executed to various addresses in
                                ;       each available memory 4k word block. The various addresses
                                ;       are determined by floating a 1 then a 0 through the address bits.
                                ;
                                ; TEST STEPS:
                                ;
                                ;    BEGIN
                                ;       Write to TSSR to soft initialize
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                                ;
```

TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

```
                                ;       REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
                                ;         BEGIN
                                ;               Get a valid modulo-4 test address
                                ;               Do a GET STATUS command from the test address
                                ;         END
                                ;    END
                                ;--

                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                        T3.1:
026776  104402                                                                            TRAP    C$BSUB

                                ;Write to TSSR to soft initialize
                                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
                                        BCS     15$                     ;BR IF SOFT INIT = OK
                                        NEXT.ERRNO
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
026010  104455                                                                         TRAP    C$ERDF
026012  000455                                                                         .WORD   301
026014  003652                                                                         .WORD   SFIERR
026016  012034                                                                         .WORD   SFIMSG

                                ;Do a WRITE CHARACTERISTICS to setup a message buffer
                                15$:
                                        MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
                                        JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
                                        CLR     KTENABLE                ;TURN OFF KT-11
                                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
                                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                                        FORCERROR       17$
                                        BCS     20$                     ;BR IF SSR SET IN CHKTSSR
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        NEXT.ERRNO
                                17$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
026064  104455                                                                          TRAP    C$ERDF
026066  000456                                                                         .WORD   302
026070  030122                                                                         .WORD   T12WRTSSR
026072  012046                                                                         .WORD   PKTSSR

                                ;Verify a Get Status can be fetched from each address
                                ;Get a valid modulo-4 test address
                                ;Do a GET STATUS command from the test address
                                20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                                        CLR     T12KT                   ;TEST ABOVE 28K SWITCH
                                        MOV     #T12BLK,R2              ;POINT TO TEST PATTERN TABLE
                                T121LOOP:
                                        CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
                                        MOV     (R2)+,R1                ;GET TEST PATTERN ADDRESS
                                        CLR     R0                      ;ASSUME NO TEST ABOVE 28K
                                        TST     T12KT                   ;TEST ABOVE 28K THIS TIME?
                                        BEQ     25$                     ;BR IF NO
                                        MOV     -2(R2),R0               ;GET TEST PATTERN AGAIN
                                        BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
                                        MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
                                25$:    JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
```

TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

```
4529 026150  103034                        BCC     65$                     ;BR IF INVALID PACKET ADDRESS
4530 026152  013704  027654                MOV     T12LOADD,R4             ;COPY CURRENT PACKET LOW ADDRESS
4531 026156  013703  027652                MOV     T12HIADD,R3             ;COPY CURRENT PACKET HIGH ADDRESS
4532 026162  004737  031236                JSR     PC,T12SETGET            ;SETUP CURRENT PACKET TO GET STATUS
4533 026166  042703  177774                BIC     #^C<A1716>,R3           ;SAVE ADDRESS BITS 17+16
4534 026172  050304                        BIS     R3,R4                   ;SETUP 18 BIT PACKET ADDRESS
4535 026174  004737  017274                JSR     PC,KTOFF                ;TURN OFF KT-11
4536 026200  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4537 026204  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4538 026210                                FORCERROR       32$
4539 026224  103405                        BCS     40$                     ;BR IF SSR SET IN CHKTSSR
4540 026226  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4541 026230                                NEXT.ERRNO
4542 026230                        32$:    ERRDF   ERRNO,T12GETSSR,PKTGETS ;DEVICE FATAL SSR FAILED TO SET
     026230  104455                                                                TRAP    C$ERDF
     026232  000457                                                                .WORD   303
     026234  030046                                                                .WORD   T12GETSSR
     026236  012064                                                                .WORD   PKTGETS
4543 026240                        40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     026240  104406                                                                TRAP    C$CLP1
4544 026242                        65$:
4545 026242                                FORCEXIT        80$
4546 026252  020227  030016                CMP     R2,#T12TBE              ;DONE ALL TSTBLK TEST PATTERNS?
4547 026256  103002                        BHIS    70$                     ;BR IF YES
4548 026260  000137  026110                JMP     T121LOOP                ;DO ANOTHER MODULO- 4 ADDRESS
4549 026264  005737  027660        70$:    TST     T12KT                   ;DONE ABOVE 28K TESTING TOO?
4550 026270  003012                        BGT     80$                     ;BR IF YES
4551 026272  005737  003132                TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
4552 026276  001407                        BEQ     80$                     ;BR IF NO
4553 026300  012737  000001  027660        MOV     #1,T12KT                ;SET SWITCH
4554 026306  012702  027664                MOV     #T12BLK,R2              ;RESET TEST PATTERN TABLE
4555 026312  000137  026110                JMP     T121LOOP                ;DO ABOVE 28K TESTING
4556 026316  004737  017274        80$:    JSR     PC,KTOFF                ;TURN OFF KT11
4557 026322                                ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     026322                                                                     L10043:
     026322  104403                                                                TRAP    C$ESUB
4558 026324  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
4559 026330  001402                        BEQ     100$                    ;BRANCH IF NOT
4560 026332  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4561 026336                        100$:
4562
4563                                       .SBTTL  TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS
4564                               ;++
4565                               ; TEST 3: SUBTEST 2:
4566                               ;
4567                               ; SUBTEST DESCRIPTION:
4568                               ;
4569                               ;       This subtest verifies the controller can deposit message packets
4570                               ;       to all available memory locations.
4571                               ;       Write Characteristics commands are executed with message
4572                               ;       buffer addresses set to various addresses in each available
4573                               ;       memory location.
4574                               ;       The various addresses are determined by floating a 1 then a 0
4575                               ;       through the address bits.
4576                               ;
4577                               ; TEST STEPS:
4578                               ;
```

TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

```
                                ;    BEGIN
                                ;       Write to TSSR to soft initialize
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer to compare
                                ;
                                ;       REPEAT FOR SELECTED ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
                                ;         BEGIN
                                ;               Get a valid modulo-4 test address
                                ;               Set the packet message buffer to the TEST ADDRESS
                                ;               Do a WRITE CHARACTERISTICS
                                ;               Restore the test message buffer to background pattern
                                ;         END
                                ;    END
                                ;--

026336                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
026336                                                                                  T3.2:
026336  104402                                                                          TRAP    C$BSUB

                                ;Write to TSSR to soft initialize
026340  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
026344  103405                          BCS     15$                     ;BR IF SOFT INIT = OK
026346                                  NEXT.ERRNO
026346  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
026350                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
026350  104455                                                                          TRAP    C$ERDF
026352  000460                                                                          .WORD   304
026354  003652                                                                          .WORD   SFIERR
026356  012034                                                                          .WORD   SFIMSG

                                ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
026360                          15$:
026360  012704  027610                  MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
026364  004737  031170                  JSR     PC,T12SWRT              ;SET PACKET TO WRITE CHARACTERISTICS
026370  004737  017274                  JSR     PC,KTOFF                ;TURN OFF KT-11
026374  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
026400  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
026404                                  FORCERROR       17$
026420  103405                          BCS     20$                     ;BR IF SSR SET IN CHKTSSR
026422  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
026424                                  NEXT.ERRNO
026424                          17$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
026424  104455                                                                          TRAP    C$ERDF
026426  000461                                                                          .WORD   305
026430  030122                                                                          .WORD   T12WRTSSR
026432  012046                                                                          .WORD   PKTSSR

                                ;Get a valid modulo-4 test address
                                ;Set the packet message buffer to the test address
                                ;Do a WRITE CHARACTERISTICS
026434  005037  002222          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
026440  012703  027664                  MOV     #T12BLK,R3              ;POINT TO TEST PATTERN TABLE
026444                          T122LOOP:
026444  012301                          MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
026446  010100                          MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
026450  042700  177774                  BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
026454  042701  000003                  BIC     #3,R1                   ;GET RID OF A17 AND A16
```

TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

```
4626 026460  004737  030666                    JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
4627 026464  103402                            BCS     25$                     ;BR IF VALID MESSAGE BUFFER ADDRESS
4628 026466  000137  026564                    JMP     150$                    ;GET ANOTHER TEST PATTERN TO TRY
4629 026472  012704  027610          25$:      MOV     #T12PACKET,R4           ;SET THE COMMAND PACKET ADDRESS
4630 026476  004737  031170                    JSR     PC,T12SWRT              ;SETUP T12PACKET TO WRITE CHAR.
4631 026502  013737  027654  027620            MOV     T12LOADD,T12DATA        ;SETUP LOW ORDER MESSAGE BUFFER ADD.
4632 026510  013737  027652  027622            MOV     T12HIADD,T12DATA+2      ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
4633 026516  004737  017274                    JSR     PC,KTOFF                ;TURN OFF KT-11
4634 026522  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4635 026526  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4636 026532                                    FORCERROR       32$
4637 026546  103405                            BCS     50$                     ;BR IF SSR SET IN CHKTSSR
4638 026550  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4639 026552                                    NEXT.ERRNO
4640 026552                          32$:      ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     026552  104455                                                                    TRAP    C$ERDF
     026554  000462                                                                    .WORD   306
     026556  030122                                                                    .WORD   T12WRTSSR
     026560  012046                                                                    .WORD   PKTSSR
4641 026562                          50$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     026562  104406                                                                    TRAP    C$CLP1
4642 026564                          150$:
4643 026564                                    FORCEXIT        160$
4644 026574  020327  030016                    CMP     R3,#T12TBE              ;DONE ALL TST12BLK TEST PATTERN ?
4645 026600  103002                            BHIS    160$                    ;BR IF YES
4646 026602  000137  026444                    JMP     T122LOOP                ;DO ANOTHER MODULO- 4 ADDRESS
4647 026606  004737  017274          160$:     JSR     PC,KTOFF                ;TURN OFF KT11
4648 026612                                    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     026612                                                                         L10044:
     026612  104403                                                                    TRAP    C$ESUB
4649 026614  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
4650 026620  001402                            BEQ     180$                    ;BRANCH IF NOT
4651 026622  004737  017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4652 026626                          180$:
4653
4654
4655
4656                                           .SBTTL  TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS
4657                                   ;**
4658                                   ; TEST 3: SUBTEST 3:
4659                                   ;
4660                                   ; SUBTEST DESCRIPTION:
4661                                   ;
4662                                   ;       This subtest verifies the controller can fetch a
4663                                   ;       Write Characteristics data block from all available
4664                                   ;       memory locations.
4665                                   ;       Write Characteristics commands are executed with
4666                                   ;       characteristic data blocks at various memory addresses.
4667                                   ;       The various memory addresses are determined by floating
4668                                   ;       a 1 then a 0 through the address bits.
4669                                   ;
4670                                   ; TEST STEPS:
4671                                   ;
4672                                   ;    BEGIN
4673                                   ;       Write to TSSR to soft initialize
4674                                   ;
4675                                   ;       REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
```

TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

```
                                ;          BEGIN
                                ;               Get a valid test address
                                ;               Set the test packet characteristics data pointer to the
                                ;                 test address.
                                ;               Store expected characteristic data in test address block
                                ;               Do a WRITE CHARACTERISTIC command
                                ;          END
                                ;     END
                                ;--

                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////
026626                                                                                      T3.3:
026626  104402                                                                              TRAP    C$BSUB

                                ;Write to TSSR to soft initialize
026630  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
026634  103405                          BCS     20$                     ;BR IF SOFT INIT = OK
026636                                  NEXT.ERRNO
026636  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
026640                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
026640  104455                                                                              TRAP    C$ERDF
026642  000463                                                                              .WORD   307
026644  003652                                                                              .WORD   SFIERR
026646  012034                                                                              .WORD   SFIMSG

                                ;Get a valid test address
026650  005037  002222          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
026654  005037  027660                  CLR     T12KT                   ;TEST ABOVE 28K SWITCH
026660  012703  027664                  MOV     #T12BLK,R3              ;POINT TO TEST PATTERN TABLE
026664                          T123LOOP:
026664  005037  003134                  CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
026670  012301                          MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
026672  010100                          MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
026674  042700  177774                  BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
026700  042701  000003                  BIC     #3,R1                   ;GET RID OF A17 AND A16
026704  005737  027660                  TST     T12KT                   ;TEST ABOVE 28K THIS TIME?
026710  001407                          BEQ     25$                     ;BR IF NO
026712  016300  177776                  MOV     -2(R3),R0               ;GET TEST PATTERN AGAIN
026716  042700  177774                  BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
026722  012737  000001  003134          MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
026730  004737  030666          25$:    JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
026734  103402                          BCS     30$                     ;BR IF VALID TEST ADDRESS
026736  000137  027040                  JMP     60$                     ;GET NEXT TEST PATTERN
                                ;Set the test packet characteristics data pointer to the test address
026742  012704  027610          30$:    MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
026746  004737  031170                  JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
026752  013764  027654  000002          MOV     T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
026760  013764  027652  000004          MOV     T12HIADD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS
026766  004737  031300                  JSR     PC,T12CHAR              ;STORE EXPECTED DATA IN DATA BLOCK
                                ;Do a WRITE CHARACTERISTIC command
026772  004737  017274                  JSR     PC,KTOFF                ;TURN OFF KT-11
026776  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
027002  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
027006                                  FORCERROR       32$
027022  103405                          BCS     40$                     ;BR IF SSR SET IN CHKTSSR
027024  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

```
                                        NEXT.ERRNO
                              32$:      ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
                                                                              TRAP    C$ERDF
                                                                              .WORD   308
                                                                              .WORD   T12WRTSSR
                                                                              .WORD   PKTSSR
                              40$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                              TRAP    C$CLP1
                              60$:
                                        CMP     R3,#T12TBE              ;DONE ALL TSTBLK TEST PATTERNS?
                                        BHIS    65$                     ;BR IF YES
                                        JMP     T123LOOP                ;DO ANOTHER MODULO- 4 ADDRESS
                              65$:      TST     T12KT                   ;DONE ABOVE 28K TESTING TOO?
                                        BGT     70$                     ;BR IF YES
                                        TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
                                        BEQ     70$                     ;BR IF NO
                                        MOV     #1,T12KT                ;SET SWITCH
                                        MOV     #T12BLK,R3              ;RESET TEST PATTERN TABLE
                                        JMP     T123LOOP                ;DO ABOVE 28K TESTING
                              70$:      JSR     PC,KTOFF                ;TURN OFF KT11
                                        ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\
                                                                              L10045:
                                                                              TRAP    C$ESUB
                                        TST     FATFLG                  ;ANY FATAL ERRORS ?
                                        BEQ     75$                     ;BRANCH IF NOT
                                        JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
                              75$:

                                .SBTTL  TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES
                              ;**
                              ; TEST 3: SUBTEST 4:
                              ;
                              ; SUBTEST DESCRIPTION:
                              ;
                              ;       This subtest verifies the NXM error bit in the TSSR
                              ;       register is set when attempting to fetch data (a characteristic
                              ;       data block) from selected nonexistent locations.
                              ;       If NXM fails to set it is likely that an LSI 11 Bus drivar is
                              ;       failing to assert an address line.
                              ;       Addresses tested include all combinations of high-order address
                              ;       bits (i.e bits 16-21).
                              ;       ************************************************************
                              ;                               CAUTION
                              ;
                              ;               The LSI BUS drivers for all available address lines(16-21)
                              ;       are only checked when running on a 11/23B system with more than
                              ;       128K words of memory!
                              ;       ************************************************************
                              ;
                              ; TEST STEPS:
                              ;
                              ;    BEGIN
                              ;       Write to TSSR to soft initialize
                              ;       Do a write characteristic command
                              ;       Invert the extended features switch
                              ;
                              ;       REPEAT FOR SELECTED NON-EXISTENT MEMORY ADDRESSES
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
                                        ;          BEGIN
                                        ;               Get an invalid test address
                                        ;               Set the test packet characteristics data pointer to the
                                        ;                 test address.
                                        ;               Do a WRITE CHARACTERISTIC command
                                        ;               If TSSR register NXM bit not set then print error message
                                        ;          END
                                        ;     END
                                        ;--

                                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                                        T3.4:
027124  104402                                                                                                TRAP    C$BSUB

027126  005737  003144                          TST     T23A                    ;26-APR-83 REV B - CHK FOR 23A CPU
027132  001406                                  BEQ     5$                      ;26-APR-83 REV B - BR, IF NOT 23A
027134  023727  002120  007777                  CMP     L$HIME,#7777            ;26-APR-83 REV B - CHK FOR > 256KB
027142  103402                                  BLO     5$                      ;26-APR-83 REV B - BR, IF < 256KB
027144  000137  027536                          JMP     NOEXTF                  ;26-APR-83 REV B - JMP OVER 256KB
027150                                  5$:
027150  005737  003136                          TST     NXMFLG                  ;GOT ENOUGH MEMORY?
027154  001002                                  BNE     10$                     ;IF SET STAY
027156  000137  027536                          JMP     NOEXTF                  ;LEAVE IF NOT SET

                                        ;Write to TSSR to soft initialize

027162  004737  015774                  10$:    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
027166  103405                                  BCS     11$                     ;BR IF SOFT INIT = OK
027170                                          NEXT.ERRNO
027170  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027172                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
027172  104455                                                                                          TRAP    C$ERDF
027174  000465                                                                                          .WORD   309
027176  003652                                                                                          .WORD   SFIERR
027200  012034                                                                                          .WORD   SFIMSG

                                        ;Do a WRITE CHARACTERISTIC command so to invert switch

027202                                  11$:    CKLOOP                          ;LOOP IF SELECTED
027202  104406                                                                                          TRAP    C$CLP1
027204  012704  027610                          MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
027210  004737  031170                          JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
027214  005037  003134                          CLR     KTENABLE                ;TURN OFF KT-11
027220  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
027224  004737  016336                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
027230                                          FORCERROR       15$
027244  103405                                  BCS     17$                     ;BR IF SSR SET IN CHKTSSR
027246  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027250                                          NEXT.ERRNO
027250                                  15$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
027250  104455                                                                                          TRAP    C$ERDF
027252  000466                                                                                          .WORD   310
027254  030122                                                                                          .WORD   T12WRTSSR
027256  012046                                                                                          .WORD   PKTSSR
027260                                  17$:    CKLOOP                          ;LOOP IF SELECTED
027260  104406                                                                                          TRAP    C$CLP1
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
4822 027262  004737  020444                    JSR     PC,INVERT               ;INVERT THE SWITCH
4823
4824                                ;Get an invalid test address
4825
4826 027266  005037  002222          20$:      CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
4827 027272                         25$:
4828 027272  013737  003142  027652            MOV     NXMHI,T12HIADD          ;SAVE TEST ADDRESS HIGH
4829 027300  013737  003140  027654            MOV     NXMLO,T12LOADD          ;SAVE TEST ADDRESS LOW
4830 027306                         T124LOOP:
4831
4832                                ;Set the test packet characteristics data pointer to the
4833                                ;  test address.
4834
4835 027306  012704  027610          30$:      MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
4836 027312  004737  031170                    JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
4837 027316  013764  027654  000002            MOV     T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
4838 027324  013764  027652  000004            MOV     T12HIADD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS
4839
4840                                ;Do a WRITE CHARACTERISTIC command
4841 027332  004737  017274                    JSR     PC,KTOFF                ;TURN OFF KT-11
4842 027336  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4843 027342  004737  016250                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
4844 027346                                    FORCERROR       32$
4845 027362  103407                            BCS     40$                     ;BR IF SSR SET IN CHKTSSR
4846 027364  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4847 027366                                    NEXT.ERRNO
4848 027366                         32$:      ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     027366  104455                                                                    TRAP    C$ERDF
     027370  000467                                                                    .WORD   311
     027372  030122                                                                    .WORD   T12WRTSSR
     027374  012046                                                                    .WORD   PKTSSR
4849 027376  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4850 027402                         40$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027402  104406                                                                    TRAP    C$CLP1
4851 027404                                    FORCERROR       45$,NOTSSR
4852 027414                                    ESCAPE  SUB                     ;BY PASS SUBTEST IF FATAL ERROR
     027414  104410                                                                    TRAP    C$ESCAPE
     027416  000124                                                                    .WORD   L10046-.
4853                                ;If TSSR register NXM bit not set then print error message
4854 027420                         45$:
4855 027420  016501  000002                    MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
4856 027424                                    FORCERROR       52$
4857 027440  032701  004000                    BIT     #NXM,R1                 ;NXM SET?
4858 027444  001012                            BNE     60$                     ;BR IF YES
4859 027446                                    NEXT.ERRNO
4860 027446  013737  027654  002240  52$:      MOV     T12LOADD,ERRLO          ;MEMORY TEST ADDRESS LOW
4861 027454  013737  027652  002236            MOV     T12HIADD,ERRHI          ;MEMORY TEST ADDRESS HIGH
4862 027462                                    ERRHRD  ERRNO,T12NXM,ADDSSR     ;REPORT ADDRESS AND TSSR ERROR
     027462  104456                                                                    TRAP    C$ERHRD
     027464  000470                                                                    .WORD   312
     027466  030557                                                                    .WORD   T12NXM
     027470  012126                                                                    .WORD   ADDSSR
4863
4864 027472                         60$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027472  104406                                                                    TRAP    C$CLP1
4865 027474                                    FORCEXIT        90$
4866 027504  005737  003144                    TST     T23A                    ;IS IT A 11/23A?
```

```
4867 027516  001012                    BNE     90$                 ;YES WERE DONE
4868 027512  013700  027652            MOV     T12HIADD,R0         ;GET CURRENT HIGH ADDRESS
4869 027516  005200              65$:  INC     R0                  ;GET NEXT ADDRESS
4870 027520  020027  000077            CMP     R0,#77              ;DONE A21-A16?
4871 027524  101004                    BHI     90$                 ;BR IF YES
4872 027526  010037  027652      75$:  MOV     R0,T12HIADD         ;SETUP NEW HIGH ORDER ADDRESS
4873 027532  000137  027306            JMP     T124LOOP            ;DO ANOTHER NON-EXISTENT ADDRESS
4874 027536                      90$:
4875 027536                      NOEXTF:
4876 027536  004737  017274            JSR     PC,KTOFF            ;TURN OFF KT11
4877 027542                            ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\
     027542                                                                L10046:
     027542  104403                                                            TRAP    C$ESUB
4878 027544  005737  002222            TST     FATFLG              ;ANY FATAL ERRORS ?
4879 027550  001402                    BEQ     100$                ;BRANCH IF NOT
4880 027552  004737  017202            JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
4881 027556  004737  016456      100$: JSR     PC,TSTLOOP          ;SHOULD WE DO ITERATIONS?
4882 027562  103002                    BCC     105$                ;BR IF NO
4883 027564  000137  025776            JMP     T12LOOP             ;LOOP UNTIL ITERATION COUNT DONE
4884 027570                      105$:
4885 027570  004737  017274            JSR     PC,KTOFF            ;TURN OFF MEMORY MANAGEMENT
4886 027574  005037  003150            CLR     T3BFLG              ;CLEAR TEST FLAG
4887 027600                            EXIT    TST                 ;ALL DONE THIS TEST
     027600  104432                                                            TRAP    C$EXIT
     027602  001540                                                            .WORD   L10042-.
4888
4889
4890                                   ;*
4891                                   ;LOCAL STORAGE FOR THIS TEST
4892                                   ;-
4893
4895         027610                            .=<.+10>&177770
4897 027610                      T12PACKET:                        ;COMMAND PACKET FOR TEST
4898 027610  100004                    .WORD   100004              ;WRITE CHARACTERISTICS COMMAND, WITH ACK
4899 027612  027620                    .WORD   T12DATA             ;ADDRESS OF CHARACTERISTICS BLOCK
4900 027614  000000                    .WORD   0
4901 027616  000010                    .WORD   8.                  ;STARTING VALUE OF BLOCK SIZE
4902
4903 027620                      T12DATA:                          ;CHARACTERISTICS DATA BLOCK
4904 027620  027632                    .WORD   T12BFR              ;LOW ADDRESS OF MESSAGE BUFFER
4905 027622  000000                    .WORD   0                   ;HIGH ORDER OF MESSAGE BUFFER
4906 027624  000016                    .WORD   14.                 ;LENGTH OF MESSAGE BUFFER
4907 027626  000000  000000            .WORD   0,0
4908
4909 027632                      T12BFR: .BLKW   8.                ;MESSAGE BUFFER
4910
4911 027652  000000              T12HIADD:       .WORD   0         ;HIGH ADDRESS
4912 027654  000000              T12LOADD:       .WORD   0         ;LOW ADDRESS
4913 027656  000000              T12PAR6:        .WORD   0         ;ADDRESS IN PAR FORMAT
4914 027660  000000              T12KT:          .WORD   0         ;TEST ABOVE 28K SWITCH
4915 027662  000000              T124TST:        .WORD   0         ;ADDRESS TEST BIT
4916                                   ;*
4917                                   ;
4918                                   ;TABLE OF ADDRESSES
4919                                   ;
4920                                   ;-
4921 027664  000001              T12BLK: .WORD   000001
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
4922 027666  000002                          .WORD   000002
4923 027670  000003                          .WORD   000003
4924 027672  000005                          .WORD   000005
4925 027674  000006                          .WORD   000006
4926 027676  000007                          .WORD   000007
4927 027700  000011                          .WORD   000011
4928 027702  000012                          .WORD   000012
4929 027704  000013                          .WORD   000013
4930 027706  000021                          .WORD   000021
4931 027710  000022                          .WORD   000022
4932 027712  000023                          .WORD   000023
4933 027714  000041                          .WORD   000041
4934 027716  000042                          .WORD   000042
4935 027720  000043                          .WORD   000043
4936 027722  000101                          .WORD   000101
4937 027724  000102                          .WORD   000102
4938 027726  000103                          .WORD   000103
4939 027730  000201                          .WORD   000201
4940 027732  000202                          .WORD   000202
4941 027734  000203                          .WORD   000203
4942 027736  000401                          .WORD   000401
4943 027740  000402                          .WORD   000402
4944 027742  000403                          .WORD   000403
4945 027744  001001                          .WORD   001001
4946 027746  001002                          .WORD   001002
4947 027750  001003                          .WORD   001003
4948 027752  002001                          .WORD   002001
4949 027754  002002                          .WORD   002002
4950 027756  002003                          .WORD   002003
4951 027760  004001                          .WORD   004001
4952 027762  004002                          .WORD   004002
4953 027764  004003                          .WORD   004003
4954 027766  010001                          .WORD   010001
4955 027770  010002                          .WORD   010002
4956 027772  010003                          .WORD   010003
4957 027774  020001                          .WORD   020001
4958 027776  020002                          .WORD   020002
4959 030000  020003                          .WORD   020003
4960 030002  040001                          .WORD   040001
4961 030004  040002                          .WORD   040002
4962 030006  040003                          .WORD   040003
4963 030010  100001                          .WORD   100001
4964 030012  100002                          .WORD   100002
4965 030014  100003                          .WORD   100003
4966 030016  177777                T12TBE:   .WORD           177777
4967                                         ;+
4968                                         ;LOCAL TEXT MESSAGES FOR TEST
4969                                         ;-
4970
4971 030020      104     115     101  T12TST12ID:        .ASCIZ  'DMA Memory Addressing'
4972 030046      103     157     156  T12GETSSR:         .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
4973 030122      103     157     156  T12WRTSSR:         .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
4974 030211      115     145     163  T12MSGBUF:         .ASCIZ  'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'
4975 030307      102     141     143  T12BKGND:          .ASCIZ  'Background Pattern Disturbed By WRITE CHARACTERISTICS'
4976 030375      105     170     160  T12NINT:           .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
4977 030466      127     162     151  T12DPR:            .ASCIZ  'Write Characteristic data in ram does not match expected'
4978 030557      124     123     123  T12NXM:            .ASCIZ  'TSSR NXM bit failed to set when non-existent memory address specifi
ed'
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
                                        .EVEN


                                ;+
                                ;ROUTINE TO CONVERT A TEST PATTERN TO A VALID ADDRESS IN DIAGNOSTIC FREE SPACE
                                ;       DIAGNOSTIC FREE SPACE IS BETWEEN THE END OF THE DIAGNOSTIC AND THE
                                ;       BEGINNING OF THE SUPERVISOR. THIS IS ALWAYS BELOW 24K.
                                ;       IF MEMORY ABOVE 28K SPECIFIED (VIA R1) THEN PAR 6 IS SET
                                ;       TO THE RELOCATION BASE.
                                ;
                                ; INPUTS:
                                ;
                                ;       R0      HIGH ORDER ADDRESS BITS
                                ;       R1      LOW ORDER ADDRESS BITS
                                ;
                                ; OUPTUTS:
                                ;       T12PAR6 = ADDRESS BIASED TO PAR6 IF >28K UNDER TEST
                                ;       T12HIADD = HIGH ORDER ADDRESS IN NON PAR6 FORMAT
                                ;       T12LOADD = LOW ORDER ADDRESS IN NON PAR6 FORMAT
                                ;       C BIT = 1 IF GOOD ADDRESS RETURNED
                                ;       C BIT = 0 IF TEST PATTERN DID NOT YIELD A VALID ADDRESS
                                ;-
030666                          T12CONVERT:
030666                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
030672  005037  027654                  CLR     T12LOADD        ;CLEAR LOW ADDRESS
030676  005037  027652                  CLR     T12HIADD        ;CLEAR HIGH ADDRESS
030702  005037  027656                  CLR     T12PAR6         ;CLEAR PAR6 BIASED ADDRESS
030706  042701  170000                  BIC     #^C<7777>,R1    ;FORCE TO LOWER 12 BITS OF ADDRESS
030712  010005                          MOV     R0,R5           ;SAVE HIGH ORDER ADDRESS BITS
030714  004737  017274                  JSR     PC,KTOFF        ;SHUTOFF MEMORY MANAGEMENT
030720  013702  003124                  MOV     FREE,R2         ;GET FIRST FREE ADDRESS
030724  062702  000020                  ADD     #16.,R2         ;IN CASE TEST PATTERN=0
030730  060102                          ADD     R1,R2           ;ADD IN TEST PATTERN
030732  042702  000003                  BIC     #3,R2           ;MAKE IT MODULO-4
030736  013703  003130          25$:    MOV     FREEHI,R3       ;GET LAST FREE ADDRESS
030742  162703  000020                  SUB     #16.,R3         ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
030746  010237  027654                  MOV     R2,T12LOADD     ;SAVE POSSIBLE LOW ADDRESS
030752  010237  027656                  MOV     R2,T12PAR6      ;SAVE IT IN PAR6 BIASED TOO
030756  020203                          CMP     R2,R3           ;IS THIS ADDRESS ABOVE FREE SPACE?
030760  101007                          BHI     35$             ;BR IF YES
030762  020237  003124                  CMP     R2,FREE         ;IS IT IN FREE SPACE?
030766  103007                          BHIS    50$             ;BR IF YES- ITS GOOD
030770  005737  003134                  TST     KTENABLE        ;TESTING ABOVE 28K?
030774  001004                          BNE     50$             ;BR IF YES
030776  000424                          BR      90$             ;BR IF NOT IN FREE SPACE
031000  162702  000020          35$:    SUB     #16.,R2         ;FORCE FIT THE TEST PATTERN
031004  000754                          BR      25$             ;TRY THIS TEST PATTERN ADDRESS
031006                          50$:
031006  005737  003134                  TST     KTENABLE        ;TESTING ABOVE 28K?
031012  001420                          BEQ     100$            ;BR IF NO
031014  005737  003132                  TST     KTFLG           ;ANY MEMORY ABOVE 28K?
031020  001413                          BEQ     90$             ;BR IF NO
031022  004737  017256                  JSR     PC,KTON         ;TURN ON MEMORY MANAGEMENT
031026  010500                          MOV     R5,R0           ;GET HIGH ORDER ADDRESS
031030  010037  027652                  MOV     R0,T12HIADD     ;SAVE POSSIBLE HIGH ADDRESS
031034  010201                          MOV     R2,R1           ;GET COMPUTED LOW ORDER ADDRESS
031036  004737  017316                  JSR     PC,SETMAP       ;RETURN PAR6 BIASED ADDRESS IN R0
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
5036 031042  010037  027656                    MOV     R0,T12PAR6      ;COPY PAR6 BIASED ADDRESS
5037 031046  103403                            BCS     105$            ;BR IF VALID ADDRESS
5038 031050  000241                  90$:      CLC                     ;CLR C BIT FOR FAILURE
5039 031052  000401                            BR      105$            ;
5040 031054  000261                  100$:     SEC                     ;SET SUCCESS
5041 031056  000207                  105$:     RTS     PC              ;RETURN
5042
5043
5044                                 ;+
5045                                 ;
5046                                 ;ROUTINE TO READ THE FIRST 2 BYTES FROM RAM
5047                                 ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
5048                                 ;
5049                                 ;INPUT:
5050                                 ;
5051                                 ;       R4      ADDRESS OF THE COMMAND PACKET
5052                                 ;       R5      FIRST DEVICE UNIBUS ADDRESS
5053                                 ;
5054                                 ;OUTPUT:
5055                                 ;
5056                                 ;       CARRY   SET - RAM MATCHES PACKET
5057                                 ;               CLR - RAM DOES NOT MATCH PACKET
5058                                 ;
5059                                 ;IMPLICIT OUTPUT:
5060                                 ;
5061                                 ;       THE TABLE RAMDATA IS FILLED WITH THE
5062                                 ;       DATA HELD IN RAM.
5063                                 ;       RAMSIZ  SET TO 2 FOR PRAMPKT ROUTINE
5064                                 ;
5065                                 ;SIDE EFFECTS:
5066                                 ;
5067                                 ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
5068                                 ;
5069                                 ;-
5070 031060                         T12CKRAM::
5071 031060                                 SAVREG                          ;SAVE THE GENERAL REGISTERS
5072 031064  012701  002242                MOV     @RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
5073 031070  012702  000201                MOV     @RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
5074 031074  005003                        CLR     R3                      ;CLEAR THE ERROR FLAG
5075 031076  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR
5076 031102  112765  000000  000000        MOVB    @0,TSDB(R5)             ;SET MAINTENANCE MODE
5077 031110  004737  016336        10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5078 031114  010265  000000                MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
5079 031120  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5080 031124  116511  000000                MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
5081 031130  122124                        CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
5082 031132  001401                        BEQ     20$                     ;BRANCH IF OK
5083 031134  005203                        INC     R3                      ;SET ERROR FLAG
5084 031136  005202                20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
5085 031140  020227  000203                CMP     R2,@RMPKTBEG+2          ;DONE 2 BYTES?
5086 031144  002761                        BLT     10$                     ;BR IF NO
5087 031146  005703                        TST     R3                      ;WAS AN ERROR FOUND ?
5088 031150  001402                        BEQ     30$                     ;BRANCH IF NOT
5089 031152  000241                        CLC                             ;CLEAR CARRY TO SHOW ERROR
5090 031154  000401                        BR      50$                     ;AND EXIT
5091 031156  000261                30$:    SEC                             ;SHOW GOOD COMPARE
5092 031160  012737  000002  002302  50$:    MOV     #2,RAMSIZ               ;SETUP RAMSIZ
```

```
5093 031166  000207                        RTS     PC                      ;RETURN
5094
5095                                ;+
5096                                ;
5097                                ;ROUTINE TO SETUP PACKET TO WRITE CHARACTERISTICS
5098                                ;-
5099
5100 031170                         T12SWRT:
5101 031170                                 SAVREG                          ;SAVE THE REGISTERS
5102 031174  012701  027610                 MOV     #T12PACKET,R1           ;START OF THE PACKET
5103 031200  012721  100004                 MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
5104 031204  012721  027620                 MOV     #T12DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
5105 031210  005021                         CLR     (R1)+                   ;EXTENDED ADDRESS
5106 031212  012721  000010                 MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
5107 031216  012721  027632                 MOV     #T12BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
5108 031222  005021                         CLR     (R1)+
5109 031224  012721  000016                 MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
5110 031230  005021                         CLR     (R1)+
5111 031232  005011                         CLR     (R1)
5112 031234  000207                         RTS     PC                      ;RETURN
5113                                ;+
5114                                ;
5115                                ;ROUTINE TO SETUP A GET STATUS COMMAND PACKET AT CURRENT PACKET ADDRESS
5116                                ;
5117                                ;       R3      HIGH ORDER PACKET ADDRESS
5118                                ;       R4      LOW ORDER PACKET ADDRESS
5119                                ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG CLEAR
5120                                ;
5121                                ;-
5122
5123 031236                         T12SETGET:
5124 031236                                 SAVREG                          ;SAVE THE REGISTERS
5125 031242  010401                         MOV     R4,R1                   ;GET LOW ORDER ADDRESS
5126 031244  005737  003134                 TST     KTENABLE                ;TESTING ABOVE 28K?
5127 031250  001404                         BEQ     10$                     ;BR IF NO
5128 031252  010300                         MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
5129 031254  004737  017316                 JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
5130 031260  010001                         MOV     R0,R1                   ;GET ADDRESS
5131 031262  012700  000017         10$:    MOV     #P.GETSTATUS,R0         ;GET STATUS COMMAND CODE NO IE
5132 031266  052700  100000                 BIS     #P.ACK,R0               ;SET ACK
5133 031272  010021                         MOV     R0,(R1)+                ;STORE GET STATUS IN PACKET
5134 031274  005021                         CLR     (R1)+                   ;CLEAR UNUSED WORD
5135 031276  000207                         RTS     PC                      ;RETURN
5136
5137
5138                                ;+
5139                                ;
5140                                ;ROUTINE TO SETUP A CHARACTERISTIC DATA BLOCK AT A TEST ADDRESS
5141                                ;
5142                                ;-
5143 031300                         T12CHAR:
5144 031300                                 SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
5145 031304  012700  027620                 MOV     #T12DATA,R0             ;GET T12PACKET DATA POINTER
5146 031310  013701  027654                 MOV     T12LOADD,R1             ;ASSUME NOT ABOVE 28K
5147 031314  005737  003134                 TST     KTENABLE                ;TESTING ABOVE 28K?
5148 031320  001402                         BEQ     10$                     ;BR IF NO
5149 031322  013701  027656                 MOV     T12PAR6,R1              ;SET TEST ADDRESS ABOVE 28K
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
 5150 031326  012021                10$:    MOV     (R0)+,(R1)+     ;STORE DATA WORD 1
 5151 031330  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 2
 5152 031332  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 3
 5153 031334  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 4
 5154 031336  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 5
 5155 031340  000207                        RTS     PC              ;RETURN
 5156
 5157 031342                                ENDTST
      031342                                                                L10042:
      031342  104401                                                               TRAP    C$ETST
```

```
                                        .SBTTL  TEST  4: RAM EXERCISER TEST
                                ;+
                                ;
                                ;THIS TEST USES THE READ AND WRITE RAM (BOTH SINGLE AND 256
                                ;LOCATIONS) SELECT CODES OF THE WRITE SUBSYSTEM MEMORY COMMAND
                                ;TO EXERCISE THE CONTROLLER'S RAM MEMORY AND DMA LOGIC
                                ;
                                ;-
031344                                  BGNTST
031344                                                                          T4::

005737  002214                  TST     TSTCNT                  ;CHECK FOR RUN MODE
001402                          BEQ     10$                     ;BR, IF NOT ONLY PROGRAM RUN
005237  003400                  INC     SKIPT                   ;SET SKIP SW
012700  034003          10$:    MOV     #TST15ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
C12737  000005  002216          MOV     #5,LOOPCNT              ;PERFORM 5 ITERATIONS
031374                  T15LOOP:
                                ;
                                ;
                                ;TEST 4, SUBTEST 1
                                ;
                                ;       THIS SUBTEST WRITES THE ADDRESS (8 BITS) INTO THE
                                ;       RAM MEMORY SINGLE WORD (8 BITS) MODE
                                ;
                                ;-
031374                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
031374                                                                          T4.1:
031374  104402                                                                  TRAP    C$BSUB
031376                          SETPRI  #PRIOO                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
031376  012700  000000                                                          MOV     #PRIOO,R0
031402  104441                                                                  TRAP    C$SPRI
005737  003400                  TST     SKIPT                   ;SHOULD WE SKIP THIS SUBTEST
001402                          BEQ     10$                     ;BR, IF NOW SKIP REQUIRED
000137  031674                  JMP     50$                     ;SKIP SUBTEST
004737  034022          10$:    JSR     PC,T15REST              ;SET COMMAND PACKET
004737  034074                  JSR     PC,T15RT2               ;SET UP OTHER COMMAND PACKET
004737  015774                  JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
103405                          BCS     20$                     ;BR IF INIT WAS OK
010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
031436                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
031436  104455                                                                  TRAP    C$ERDF
031440  000621                                                                  .WORD   401
031442  003652                                                                  .WORD   SFIERR
031444  012034                                                                  .WORD   SFIMSG
031446                  20$:
012704  032720                  MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
103405                          BCS     23$                     ;BR, IF COMMAND ISSUED OK
010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
031462                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
031462  104456                                                                  TRAP    C$ERHRD
031464  000622                                                                  .WORD   402
031466  005056                                                                  .WORD   WRTMSG
031470  012034                                                                  .WORD   SFIMSG
```

TEST 4: RAM EXERCISER TEST

```
5213 031472  012703  000400          23$:   MOV     #256.,R3              ;STARTING ADDRESS FOR RAM WRITE
5214 031476  112737  000001  033431         MOVB    #1,T15BS1             ;SIZE OF TRANSFER
5215 031504  112737  000002  033430         MOVB    #2,T15BS0             ;WRITE RAM "COMMAND"
5216 031512                          25$:
5217 031512  010337  033432                 MOV     R3,T15S2              ;ADDRESS FOR RAM
5218 031516  012704  033420                 MOV     #T15PK2,R4            ;WRITE SUBSYS MEM PACKET
5219 031522  110337  033434                 MOVB    R3,T15S3              ;DATA FOR WRITE (ADDRESS)
5220 031526  010465  000000                 MOV     R4,TSDB(R5)           ;ISSUE COMMAND
5221 031532  004737  016336                 JSR     PC,CHKTSSR            ;WAIT FOR SSR
5222 031536  103407                        BCS     30$                   ;BR, IF NO ERROR
5223 031540  010001                        MOV     R0,R1                 ;ERROR, SAVE TSSR
5227 031542                                 ERRHRD  ERRNO,T15SSR,PKTSSR   ;TSSR NOT CORRECT AFTER WRITE SUB MEM
     031542  104456                                                              TRAP    C$ERHRD
     031544  000623                                                              .WORD   403
     031546  033436                                                              .WORD   T15SSR
     031550  012046                                                              .WORD   PKTSSR
5228 031552                                 ESCAPE  SUB                   ;DON'T CONTINUE IF ERROR ON WRITE
     031552  104410                                                              TRAP    C$ESCAPE
     031554  000122                                                              .WORD   L10050-.
5229 031556                          30$:   CKLOOP                        ;SCOPE LOOP
     031556  104406                                                              TRAP    C$CLP1
5230
5231
5232 031560  005203                        INC     R3                    ;NEXT ADDRESS
5233 031562  020327  010000                 CMP     R3,#10000             ;END OF RAM MEMORY CHECK
5234 031566  001351                        BNE     25$                   ;LOOP TILL ALL RAM WRITTEN
5235 031570  005002                        CLR     R2                    ;CLEAR OUT R2 HIGH BITS
5236 031572  005303                        DEC     R3                    ;SET BACK TO 7777
5237 031574  110337  033434          40$:   MOVB    R3,T15S3              ;GET DATA PATTERN BACK IN SHAPE
5238 031600  010337  033432                 MOV     R3,T15S2              ;ADDRESS FOR RAM READ
5239 031604  112737  000001  033430         MOVB    #1,T15BS0             ;READ RAM COMMAND
5240 031612  010465  000000                 MOV     R1,TSDB(R5)           ;SEND OUT PACKET ADDRESS TO CONTR.
5241 031616  004737  016336                 JSR     PC,CHKTSSR            ;WAIT FOR READY, NON-AMBIGUOUS
5242 031622  103405                        BCS     43$                   ;BR, IF NO PROBLEM
5243 031624  010001                        MOV     R0,R1                 ;SAVE TSSR
5247 031626                                 ERRDF   ERRNO,T15SSR,PKTSSR   ;TSSR NOT CORRECT
     031626  104455                                                              TRAP    C$ERDF
     031630  000624                                                              .WORD   404
     031632  033436                                                              .WORD   T15SSR
     031634  012046                                                              .WORD   PKTSSR
5248 031636                          43$:   CKLOOP                        ;SCOPE LOOP
     031636  104406                                                              TRAP    C$CLP1
5249 031640  013701  032762                 MOV     T15BFR+20,R1          ;GET RAM READ DATA
5250 031644  010302                        MOV     R3,R2                 ;SET UP FOR COMPARE
5251 031646  120102                        CMPB    R1,R2                 ;CHECK WITH DATA WRITTEN
5252 031650  001404                        BEQ     45$                   ;BR IF OK, DATA IN = DATA OUT
5256 031652                                 ERRHRD  ERRNO,T15AM4,EXPBREC  ;WRITTEN DATA NOT = TO READ
     031652  104456                                                              TRAP    C$ERHRD
     031654  000625                                                              .WORD   405
     031656  033715                                                              .WORD   T15AM4
     031660  015502                                                              .WORD   EXPBREC
5257 031662                          45$:   CKLOOP                        ;SCOPE LOOP
     031662  104406                                                              TRAP    C$CLP1
5258 031664  005303                        DEC     R3                    ;DROP DATA COUNTER (PATTERN)
5259 031666  020327  000377                 CMP     R3,#255.              ;AT BOTTOM YET
5260 031672  001340                        BNE     40$                   ;BR, IF MORE TO CHECK
5261 031674                          50$:   CKLOOP                        ;SCOPE LOOP
```

TEST 4: RAM EXERCISER TEST

```
                                                                                    TRAP    C$CLP1
5262 031676                             ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     031676                                                                       L10050:
     031676  104403                                                               TRAP    C$ESUB
5263
5264
5265 031700                             BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
     031700                                                                       T4.2:
     031700  104402                                                               TRAP    C$BSUB
5266
5267                            ;*
5268                            ;
5269                            ;TEST 4, SUBTEST 2
5270                            ;
5271                            ;
5272                            ;       THIS SUBTEST WRITES RAM WITH ALL ZEROS
5273                            ;       THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
5274                            ;
5275 031702  C04737  034022             JSR     PC,T15REST              ;RESTORE PACKET FOR WRITE CHARA
5276 031706  004737  034074             JSR     PC,T15RT2               ;RESTORE PACKET FOR WRT SUB SYS MEM
5277 031712  004737  015774             JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
5278 031716  103405                     BCS     20$                     ;BR IF INIT WAS OK
5282 031720  010001                     MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
5283 031722                             ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
     031722  104455                                                               TRAP    C$ERDF
     031724  000626                                                               .WORD   406
     031726  003652                                                               .WORD   SFIERR
     031730  012034                                                               .WORD   SFIMSG
5284 031732                     20$:
5285 031732  012704  032720             MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
5286 031736  004737  010662             JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
5287 031742  103405                     BCS     25$                     ;BR, IF COMMAND ISSUED OK
5291 031744  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5292 031746                             ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     031746  104456                                                               TRAP    C$ERHRD
     031750  000627                                                               .WORD   407
     031752  005056                                                               .WORD   WRTMSG
     031754  012034                                                               .WORD   SFIMSG
5293 031756                     25$:
5294 031756  112737  000001  033431     MOVB    #1,T15BS1               ;SET SIZE OF TRANSFER 1 BYTE
5295 031764  012704  033420             MOV     #T15PK2,R4              ;SET NEW PACKET ADDRESS
5296 031770  012703  000400             MOV     #256.,R3                ;STARTING ADDRESS IN RAM
5297 031774  112737  000002  033430     MOVB    #2,T15BS0               ;WRITE RAM COMMAND
5298 032002  105037  033434             CLRB    T15S3                   ;SET DATA TO 000
5299 032006  010337  033432     30$:    MOV     R3,T15S2                ;ADDRESS TO PACKET DATA AREA
5300 032012  010465  000000             MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
5301 032016  004737  016336             JSR     PC,CHKTSSR              ;WAIT FOR SSR
5302 032022  103405                     BCS     33$                     ;BR, IF NO PROBLEM
5303 032024  010001                     MOV     R0,R1                   ;SAVE TSSR
5307 032026                             ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032026  104456                                                               TRAP    C$ERHRD
     032030  000630                                                               .WORD   408
     032032  033436                                                               .WORD   T15SSR
     032034  012046                                                               .WORD   PKTSSR
5308 032036                     33$:    CKLOOP                          ;SCOPE LOOP
     032036  104406                                                               TRAP    C$CLP1
5309
```

TEST 4: RAM EXERCISER TEST

```
5310
5311 032040  005203                    INC     R3                      ;NEXT ADDRESS
5312 032042  020327  010000             CMP     R3,#10000               ;END OF RAM MEMORY CHECK
5313 032046  001357                    BNE     30$                     ;BR, MORE RAM TO GO
5314 032050  005303              35$:  DEC     R3                      ;SET BACK TO 7777
5315 032052  005002              40$:  CLR     R2                      ;SET TO ALL ZEROS
5316 032054  112737  000001  033430    MOVB    #1,T15BS0               ;READ RAM COMMAND
5317 032062  010337  033432            MOV     R3,T15S2                ;ADDRESS TO BE READ TO PACKET DATA
5318 032066  010465  000000            MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
5319 032072  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5320 032076  103405                    BCS     41$                     ;BR, IF ALL IS WELL
5321 032100  010001                    MOV     R0,R1                   ;SAVE TSSR
5325 032102                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032102  104456                                                            TRAP    C$ERHRD
     032104  000631                                                            .WORD   409
     032106  033436                                                            .WORD   T15SSR
     032110  012046                                                            .WORD   PKTSSR
5326 032112              41$:  CKLOOP                                  ;SCOPE LOOP
     032112  104406                                                            TRAP    C$CLP1
5327 032114  013701  032762            MOV     T15BFR+20,R1            ;PICK UP READ DATA
5328 032120  120102                    CMPB    R1,R2                   ;BOTH SHOULD BE 00000000 BINARY
5329 032122  001404                    BEQ     42$                     ;BR, IF DATA IS GOOD
5333 032124                            ERRHRD  ERRNO,T15AM3,EXPBREC    ;CHARACTERISTICS DATA NOT CORRECT
     032124  104456                                                            TRAP    C$ERHRD
     032126  000632                                                            .WORD   410
     032130  033613                                                            .WORD   T15AM3
     032132  015502                                                            .WORD   EXPBREC
5334 032134              42$:  CKLOOP                                  ;SCOPE LOOPER
     032134  104406                                                            TRAP    C$CLP1
5335 032136  012702  000377            MOV     #000377,R2              ;SET ALL ONES WORD
5336 032142  112737  000002  033430    MOVB    #2,T15BS0               ;WRITE RAM COMMAND
5337 032150  112737  000377  033434    MOVB    #000377,T15S3           ;ALL ONES PATTERN
5338 032156  010465  000000            MOV     R4,TSDB(R5)             ;PASS PACKET ADDRESS TO CONTR.
5339 032162  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR
5340 032166  103405                    BCS     43$                     ;BR, IF OK (NO ERROR)
5341 032170  010001                    MOV     R0,R1                   ;SAVE TSSR
5345 032172                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032172  104456                                                            TRAP    C$ERHRD
     032174  000633                                                            .WORD   411
     032176  033436                                                            .WORD   T15SSR
     032200  012046                                                            .WORD   PKTSSR
5346 032202              43$:  CKLOOP                                  ;SCOPE LOOP
     032202  104406                                                            TRAP    C$CLP1
5347 032204  112737  000001  033430    MOVB    #1,T15BS0               ;SET UP FOR RAM READ
5348 032212  010465  000000            MOV     R4,TSDB(R5)             ;ISSUE RAM READ
5349 032216  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5350 032222  103405                    BCS     44$                     ;BR, IF OK (NO ERROR)
5351 032224  010001                    MOV     R0,R1                   ;SAVE TSSR
5355 032226                            ERRDF   ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032226  104455                                                            TRAP    C$ERDF
     032230  000634                                                            .WORD   412
     032232  033436                                                            .WORD   T15SSR
     032234  012046                                                            .WORD   PKTSSR
5356 032236  013701  032762    44$:    MOV     T15BFR+20,R1            ;PICK UP REC'D DATA
5357 032242  120102                    CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
5358 032244  001404                    BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
5362 032246                            ERRHRD  ERRNO,T15AM2,EXPBREC    ;WRITTEN DATA NOT = TO READ
```

TEST 4: RAM EXERCISER TEST

```
        032246  104456                                                          TRAP    C$ERHRD
        032250  000635                                                          .WORD   413
        032252  033512                                                          .WORD   T15AM2
        032254  015502                                                          .WORD   EXPBREC
 5363   032256                          45$:    CKLOOP                  ;SCOPE LOOP
        032256  104406                                                          TRAP    C$CLP1
 5364   032260  005303                  DEC     R3                      ;DROP RAM ADDRESS POINTER
 5365   032262  020327  000377          CMP     R3,#255.                ;AT START YET
 5366   032266  001271                  BNE     40$                     ;BR, IF MORE RAM TO CHECK
 5367
 5368   032270                          ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        032270                                                                  L10051:
        032270  104403                                                          TRAP    C$ESUB
 5369
 5370   032272                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
        032272                                                                  T4.3:
        032272  104402                                                          TRAP    C$BSUB
 5371                                   ;*
 5372                                   ;
 5373                                   ;TEST 4, SUBTEST 3
 5374                                   ;
 5375                                   ;
 5376                                   ;       THIS SUBTEST WRITES RAM WITH ALL ONES
 5377                                   ;       THEN WALKS AN ALL ZEROS WORD DOWN THROUGH MEMORY
 5378                                   ;
 5379   032274  005737  003400          TST     SKIPT                   ;CHECK RUN MODE
 5380   032300  001402                  BEQ     10$                     ;BR, IF NO SKIP
 5381   032302  000137  032676          JMP     50$                     ;SKIP SUBTEST
 5382   032306  004737  034022  10$:    JSR     PC,T15REST              ;RESTORE PACKET FOR WRITE CHARA
 5383   032312  004737  034074          JSR     PC,T15RT2               ;RESTORE PACKET FOR WRT SUB SYS MEM
 5384   032316  004737  015774          JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
 5385   032322  103405                  BCS     20$                     ;BR IF INIT WAS OK
 5389   032324  010001                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
 5390   032326                          ERROF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
        032326  104455                                                          TRAP    C$EROF
        032330  000636                                                          .WORD   414
        032332  003652                                                          .WORD   SFIERR
        032334  012034                                                          .WORD   SFIMSG
 5391   032336                  20$:
 5392   032336  012704  032720          MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
 5393   032342  004737  010662          JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 5394   032346  103405                  BCS     25$                     ;BR, IF COMMAND ISSUED OK
 5398   032350  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5399   032352                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
        032352  104456                                                          TRAP    C$ERHRD
        032354  000637                                                          .WORD   415
        032356  005056                                                          .WORD   WRTMSG
        032360  012034                                                          .WORD   SFIMSG
 5400   032362                  25$:
 5401   032362  112737  000001  033431  MOVB    #1,T15BS1               ;SET SIZE TO 1 BYTE
 5402   032370  012704  033420          MOV     #T15PK2,R4              ;SET NEW PACKET ADDRESS
 5403   032374  012703  000400          MOV     #256.,R3                ;STARTING ADDRESS IN RAM
 5404   032400  112737  000002  033430  MOVB    #2,T15BS0               ;WRITE RAM COMMAND
 5405   032406  112737  000377  033434  MOVB    #377,T15S3              ;SET DATA TO 377
 5406   032414  010337  033432  30$:    MOV     R3,T15S2                ;ADDRESS TO PACKET DATA AREA
 5407   032420  010465  000000          MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
 5408   032424  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR
```

TEST 4: RAM EXERCISER TEST

```
5409 032430  103405                    BCS     33$                     ;BR, IF NO PROBLEM
5410 032432  010001                    MOV     R0,R1                   ;SAVE TSSR
5414 032434                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032434  104456                                                            TRAP    C$ERHRD
     032436  000640                                                            .WORD   416
     032440  033436                                                            .WORD   T15SSR
     032442  012046                                                            .WORD   PKTSSR
5415 032444                    33$:    CKLOOP                          ;SCOPE LOOP
     032444  104406                                                            TRAP    C$CLP1
5416
5417
5418 032446  005203                    INC     R3                      ;NEXT ADDRESS
5419 032450  020327  010000            CMP     R3,#10000               ;END OF RAM MEMORY CHECK
5420 032454  001357                    BNE     30$                     ;BR, MORE RAM TO GO
5421 032456  005303                    DEC     R3                      ;SET BACK TO 7777
5422 032460  112702  000377    35$:    MOVB    #377,R2                 ;SET TO ALL ONES
5423 032464  112737  000001  033430    MOVB    #1,T15BSO               ;READ RAM COMMAND
5424 032472  010337  033432            MOV     R3,T15S2                ;ADDRESS TO BE READ TO PACKET DATA
5425 032476  C10465  000000            MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
5426 032502  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5427 032506  103405                    BCS     41$                     ;BR, IF ALL IS WELL
5428 032510  010001                    MOV     R0,R1                   ;SAVE TSSR
5432 032512                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032512  104456                                                            TRAP    C$ERHRD
     032514  000641                                                            .WORD   417
     032516  033436                                                            .WORD   T15SSR
     032520  012046                                                            .WORD   PKTSSR
5433 032522                    41$:    CKLOOP                          ;SCOPE LOOP
     032522  104406                                                            TRAP    C$CLP1
5434 032524  013701  032762            MOV     T15BFR+20,R1            ;PICK UP READ DATA
5435 032530  120102                    CMPB    R1,R2                   ;BOTH SHOULD BE 11111111 BINARY
5436 032532  001404                    BEQ     42$                     ;BR, IF DATA IS GOOD
5440 032534                            ERRHRD  ERRNO,T15AM3,EXPBREC    ;CHARACTERISTICS DATA NOT CORRECT
     032534  104456                                                            TRAP    C$ERHRD
     032536  000642                                                            .WORD   418
     032540  033613                                                            .WORD   T15AM3
     032542  015502                                                            .WORD   EXPBREC
5441 032544  012702  000377    42$:    MOV     #000377,R2              ;SET ALL ONES WORD
5442 032550  012737  000002  033430    MOV     #2,T15BSO               ;WRITE RAM COMMAND
5443 032556  112737  000377  033434    MOVB    #000377,T15S3           ;ALL ONES PATTERN
5444 032564  010465  000000            MOV     R4,TSDB(R5)             ;PASS PACKET ADDRESS TO CONTR.
5445 032570  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR
5446 032574  103405                    BCS     43$                     ;BR, IF OK (NO ERROR)
5447 032576  010001                    MOV     R0,R1                   ;SAVE TSSR
5451 032600                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032600  104456                                                            TRAP    C$ERHRD
     032602  000643                                                            .WORD   419
     032604  033436                                                            .WORD   T15SSR
     032606  012046                                                            .WORD   PKTSSR
5452 032610                    43$:    CKLOOP                          ;SCOPE LOOP
     032610  104406                                                            TRAP    C$CLP1
5453 032612  112737  000001  033430            MOVB    #1,T15BSO               ;SET UP FOR RAM READ
5454 032620  010465  000000            MOV     R4,TSDB(R5)             ;ISSUE RAM READ
5455 032624  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5456 032630  103405                    BCS     44$                     ;BR, IF OK (NO ERROR)
5457 032632  010001                    MOV     R0,R1                   ;SAVE TSSR
5461 032634                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
```

TEST 4: RAM EXERCISER TEST

```
                032634  104456                                                          TRAP    C$ERHRD
                032636  000644                                                          .WORD   420
                032640  033436                                                          .WORD   T15SSR
                032642  012046                                                          .WORD   PKTSSR
   5462         032644  013701  032762          44$:    MOV     T15BFR+20,R1          ;PICK UP REC'D DATA
   5463         032650  120102                  CMPB    R1,R2                         ;CHECK WITH DATA WRITTEN
   5464         032652  001404                  BEQ     45$                           ;BR IF OK, DATA IN = DATA OUT
   5468         032654                          ERRHRD  ERRNO,T15AM2,EXPBREC          ;WRITTEN DATA NOT = TO READ
                032654  104456                                                          TRAP    C$ERHRD
                032656  000645                                                          .WORD   421
                032660  033512                                                          .WORD   T15AM2
                032662  015502                                                          .WORD   EXPBREC
   5469         032664                  45$:    CKLOOP                                ;SCOPE LOOP
                032664  104406                                                          TRAP    C$CLP1
   5470         032666  005303                  DEC     R3                            ;DROP RAM ADDRESS POINTER
   5471         032670  020327  000377          CMP     R3,#255.                      ;AT START YET
   5472         032674  001271                  BNE     40$                           ;BR. IF MORE RAM TO CHECK
   5473
   5474         032676                  50$:
   5475         032676                  ENDSUB                                        ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                032676                                                                L10052:
                032676  104403                                                          TRAP    C$ESUB
   5476
   5477         032700  004737  016456          JSR     PC,TSTLOOP                    ;DO WE NEED TO ITERATE TEST ?
   5478         032704  103002                  BCC     63$                           ;BRANCH IF NOT
   5479         032706  000137  031374          JMP     T15LOOP                       ;EXECUTE AGAIN
   5480         032712                  63$:    EXIT    TST                           ;ALL DONE THIS TEST
                032712  104432                                                          TRAP    C$EXIT
                032714  001216                                                          .WORD   L10047-.
   5481                                 ;+
   5482                                 ;LOCAL STORAGE FOR THIS TEST
   5483                                 ;
   5484
   5486                 032720                  .=<.+10>&177770
   5488         032720                  T15PACKET:                                    ;COMMAND PACKET FOR TEST
   5489         032720  100204                  .WORD   100204                        ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
   5490         032722  032730                  .WORD   T15DATA                       ;ADDRESS OF CHARACTERISTICS BLOCK
   5491         032724  000000                  .WORD   0
   5492         032726  000010                  .WORD   8.                            ;STARTING VALUE OF BLOCK SIZE
   5493         032730                  T15DATA:                                      ;CHARACTERISTICS DATA BLOCK
   5494         032730  032742                  .WORD   T15BFR                        ;ADDRESS OF MESSAGE BUFFER
   5495         032732  000000                  .WORD   0
   5496         032734  000400                  .WORD   256.                          ;LENGTH OF MESSAGE BUFFER
   5497         032736  000000  000000          .WORD   0,0
   5498         032742                  T15BFR: .BLKW   150.                          ;MESSAGE BUFFER
   5499                                 ;
   5500                                 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
   5501                                 ;
   5503                 033420                  .=<.+10>&177770
   5505         033420                  T15PK2:
   5506         033420  100206                  .WORD   100206                        ;WRITE SUB SYS MEM COMMAND, IE AND ACK
   5507         033422  033430                  .WORD   T15BF2                        ;ADDRESS OF SELECT BLOCK DATA
   5508         033424  000000                  .WORD   0
   5509         033426  000006                  .WORD   6.                            ;SIZE OF DATA PACKET
   5510
   5511                                         .EVEN
   5512         033430                  T15BF2:
```

TEST 4: RAM EXERCISER TEST

```
5513 033430     000                 T15BS0:  BYTE   0                       ;BSEL0 AREA
5514 033431     000                 T15BS1  .BYTE   0                       ;BSEL1 AREA
5515 033432  000000                 T15S2:  .WORD   0                       ;SEL 2 AREA
5516 033434  000000                 T15S3:  .WORD   0                       ;DATA AREA
5517
5518
5519
5520
5521                                ;+
5522                                ;LOCAL TEXT MESSAGES FOR TEST
5523                                ;-
5524
5525 033436     127     122     111 T15SSR: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
5526 033512     127     122     111 T15AM2: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Ones Word Read Back'
5527 033613     127     122     111 T15AM3: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Zeros Word Read Back'
5528 033715     127     122     111 T15AM4: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On Address Test'
5529 034003     122     101     115 TST15ID:        .ASCIZ  'RAM Exerciser'
5530                                        .EVEN
5531                                ;+
5532                                ;
5533                                ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5534                                ;WRITE SUBSYSTEM MEMORY COMMAND
5535                                ;
5536                                ;-
5537
5538 034022                         T15REST:
5539 034022                                 SAVREG                          ;SAVE THE REGISTERS
5540 034026  012701  032720                MOV     #T15PACKET,R1           ;START OF THE PACKET
5541 034032  012721  100204                MOV     #100204,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
5542 034036  012721  032730                MOV     #T15DATA,(R1)+          ;ADDRESS OF CHARAISTICS DATA BLOCK
5543 034042  005021                        CLR     (R1)+                   ;EXTENDED ADDRESS
5544 034044  012721  000010                MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
5545 034050  012721  032742                MOV     #T15BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
5546 034054  005021                        CLR     (R1)+
5547 034056  012721  000400                MOV     #256.,(R1)+             ;LENGTH OF MESSAGE BUFFER
5548 034062  005021                        CLR     (R1)+
5549 034064  005011                        CLR     (R1)
5550 034066  005037  032742                CLR     T15BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
5551 034072  000207                        RTS     PC                      ;RETURN
5552
5553
5554 034074                         T15RT2:
5555 034074                                 SAVREG                          ;SAVE THE REGISTERS
5556 034100  012701  033420                MOV     #T15PK2,R1              ;START OF THE PACKET
5557 034104  012721  100206                MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
5558 034110  012721  033430                MOV     #T15BF2,(R1)+           ;ADDRESS OF DATA BLOCK
5559 034114  005021                        CLR     (R1)+                   ;EXTENDED ADDRESS
5560 034116  012721  000006                MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
5561 034122  005021                        CLR     (R1)+
5562 034124  005021                        CLR     (R1)+
5563 034126  005011                        CLR     (R1)
5564 034130  000207                        RTS     PC                      ;RETURN
5565 034132                                 ENDTST
     034132                                                         L10047:
     034132  104401                                                        TRAP    C$ETST
```

```
                                        .SBTTL  TEST  5: EXTENDED FEATURES SWITCH AND TIMERS A,B
                                ;**
                                ; TEST DESCRIPTION:
                                ;
                                ;       This test verifies the Invert Extended Features function
                                ;       can logically invert the Extended features switch and
                                ;       that the internal timers A and B operate correctly.
                                ;
                                ; TEST STEPS:
                                ;
                                ;  REPEAT FOR LOOPCNT
                                ;    BEGIN
                                ;       Do Subtest 1      Verify Extended Features Switch
                                ;       Do Subtest 2    - Verify Timers A,B
                                ;    END
                                ;--


034134                                  BGNTST
034134                                                                          T5::
034134  012700  036212                    MOV     #TST16ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
034140  004737  016510                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
034144  012737  000012  002216            MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
034152                                  T16LOOP:

                                         SBTTL  TEST  5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

                                ;**
                                ; TEST 5: SUBTEST 1:
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies that the Invert Sense of Extended features
                                ;       Switch function (Write Subsystem Memory,Write Misc command)
                                ;       operates properly.
                                ;       First the state of the Extended Features switch is read in the
                                ;       message packet supplied by the write characteristics command.
                                ;       Then, the sense of the switch is logically inverted.
                                ;       A Write characteristics command is executed and it is verified
                                ;       that the Extended status register (XST4) is returned when
                                ;       in Extended mode, and not returned if not in extended mode.
                                ;       The subtest also verifies that specifying a Message Buffer
                                ;       address with any of bits 21-19 ,set will cause the command to
                                ;       be rejected.
                                ;
                                ; TEST STEPS:
                                ;
                                ;
                                ; BEGIN
                                ;       Write to TSSR register to soft initialize the controller
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                                ;       IF Extended Features Hardware Switch CLEAR
                                ;       THEN
                                ;         (* Verify Extended Features switch can be Inverted to SET *)
                                ;         Do Write Subsystem Write Miscellaneous to SET Extended Features.
                                ;         DO a WRITE CHARACTERISTICS with an extended characteristic word
                                ;         Compare the controller ram to the extended characteristic word
```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```
                                ;         If Data word in controller ram NOT= to word sent  Then Print Error
                                ;         If Message Buffer Data Length NOT= 12.  Then Print Error
                                ;       ELSE
                                ;         (* Verify Extended Features switch can be Inverted to CLEAR *)
                                ;         Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
                                ;         Do a WRITE CHARACTERISTICS without an extended characteristic word
                                ;         If Message Buffer Data Length NOT= 10.  Then Print Error
                                ;       END-IF
                                ;       (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
                                ;       Write to TSSR register to soft initialize the controller
                                ;       REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
                                ;        DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
                                ;        If TSSR termination code NOT= Function Reject  Then Print Error
                                ;       END-REPEAT
                                ; END
                                ;--
034152                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
034152                                                                                    T5.1:
034152  104402                                                                                  TRAP    C$BSUB

                                5$:
                                ;       Write to TSSR register to soft initialize the controller
034154  004737  015774                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
034160  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
034162  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
034164                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
034164  104455                                                                            TRAP    C$ERDF
034166  000764                                                                            .WORD   500
034170  003652                                                                            .WORD   SFIERR
034172  012034                                                                            .WORD   SFIMSG
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
034174  004737  037360          10$:    JSR     PC,T16REST              ;RESTORE PACKET DEFAULTS
034200  005037  002222                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
034204  012704  037540                  MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
034210  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
034214                                  FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
034230  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
034232  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
034234                                  NEXT.ERRNO
034234                          12$:    ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
034234  104455                                                                            TRAP    C$ERDF
034236  000765                                                                            .WORD   501
034240  036262                                                                            .WORD   T16SSR
034242  012046                                                                            .WORD   PKTSSR
034244  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
034250                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
034250  104406                                                                            TRAP    C$CLP1

                                ;       If Extended Features Hardware Switch Clear then:
                                ;         (* Verify Extended Features switch can be Inverted to SET *)
                                ;         REPEAT FOR TEST PATTERNS IN TSTBLK TABLE
034252  012701  037562                  MOV     #T16BFR,R1              ;MESSAGE BUFFER ADDRESS
034256  032761  000200  000012          BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH CLEAR?
034264  001402                          BEQ     20$                     ;BR IF YES
034266  000137  034636                  JMP     200$                    ;
034272  012703  002764          20$:    MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
```

```
5673                                    ;       Do Write Subsystem Write Miscellaneous to SET Extended Features.
5674
5675 034276  004737  037520                      JSR     PC,T16SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
5676 034302  012704  037610                      MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5677 034306  010465  000000                      MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5678 034312  004737  016336                      JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5679 034316                                      FORCERROR       32$                     ;@@FORCE ERROR IF FORCER=1
5680 034332  103407                              BCS     40$                     ;BR IF CARRY SET (GOOD RETURN)
5681 034334  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5682 034336                                      NEXT.ERRNO
5683 034336                              32$:    ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     034336  104455                                                                      TRAP    C$ERDF
     034340  000766                                                                      .WORD   502
     034342  036317                                                                      .WORD   T162SSR
     034344  012046                                                                      .WORD   PKTSSR
5684 034346  005237  002222                      INC     FATFLG                  ;SET FATAL ERROR FLAG
5685 034352                              40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034352  104406                                                                      TRAP    C$CLP1
5686
5687                                    ;         DO a WRITE CHARACTERISTICS with an extended characteristic word
5688 034354  012737  125252  002312              MOV     #125252,DATA            ;SETUP TEST DATA FOR EXTENDED WORD
5689 034362  012704  037540                      MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5690 034366  012764  000020  000006              MOV     #16.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
5691 034374  013737  002312  037560              MOV     DATA,T16DATA+10         ;STORE TEST DATA IN EXTENDED WORD
5692 034402  004737  010662                      JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
5693 034406                                      FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
5694 034422  103407                              BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
5695 034424  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5696 034426                                      NEXT.ERRNO
5697 034426                              42$:    ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     034426  104455                                                                      TRAP    C$ERDF
     034430  000767                                                                      .WORD   503
     034432  036262                                                                      .WORD   T16SSR
     034434  012046                                                                      .WORD   PKTSSR
5698 034436  005237  002222                      INC     FATFLG                  ;SET FATAL ERROR FLAG
5699 034442                              50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034442  104406                                                                      TRAP    C$CLP1
5700                                    ;         If the TSBA Address Register NOT= Expected  Then Print Error
5701 034444  016501  000000                      MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
5702 034450  012702  037562                      MOV     #T16BFR,R2              ;START OF THE DATA BUFFER
5703 034454  062702  000020              62$:    ADD     #16.,R2                 ;EXPECTED CONTENTS OF TSBA
5704 034460                                      FORCERROR       72$,NOTSSR              ;@@FORCE ERROR IF FORCER=1
5705 034470  020102                              CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
5706 034472  001404                              BEQ     80$                     ;ERROR IF NOT EQUAL
5707 034474                                      NEXT.ERRNO
5708 034474                              72$:    ERRHRD  ERRNO,T16TSBA,EXPREC    ;PRINT THE ERROR & EXPD/RECV
     034474  104456                                                                      TRAP    C$ERHRD
     034476  000770                                                                      .WORD   504
     034500  036430                                                                      .WORD   T16TSBA
     034502  015474                                                                      .WORD   EXPREC
5709 034504                              80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034504  104406                                                                      TRAP    C$CLP1
5710                                    ;         Compare the controller ram to the extended characteristic word
5711                                    ;         If Data word in controller ram NOT= to word sent  Then Print Error
5712 034506  012704  037550                      MOV     #T16DATA,R4             ;GET CHARACTERISTIC DATA ADDRESS
5713 034512  004737  011224                      JSR     PC,CKRAM2               ;DOES RAM DATA EQUAL DATA SENT?
5714 034516                                      FORCERROR       92$                     ;@@FORCE ERROR IF FORCER=1
```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```
5715 034532  103404                        BCS     100$                    ;BR IF YES
5716 034534                                NEXT.ERRNO
5717 034534                        92$:    ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     034534  104456                                                                TRAP    C$ERHRD
     034536  000771                                                                .WORD   505
     034540  004745                                                                .WORD   PKTRAM
     034542  015510                                                                .WORD   RAMERR
5718 034544                        100$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034544  104406                                                                TRAP    C$CLP1
5719                               ;         If Message Buffer Data Length NOT= 12.  Then Print Error
5720 034546  012702  037562                MOV     @T16BFR,R2              ;GET MESSAGE BUFFER ADDRESS
5721 034552  016201  000002                MOV     2(R2),R1                ;GET RECV DATA FIELD LENGTH
5722 034556  012702  000014                MOV     #12.,R2                 ;GET EXPD DATA FIELD LENGTH
5723 034562                                FORCERROR       112$,NOTSSR             ;@@FORCE ERROR IF FORCER=1
5724 034572  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
5725 034574  001404                        BEQ     120$                    ;ERROR IF NOT EQUAL
5726 034576                                NEXT.ERRNO
5727 034576                        112$:   ERRHRD  ERRNO,T16LEN,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     034576  104456                                                                TRAP    C$ERHRD
     034600  000772                                                                .WORD   506
     034602  036532                                                                .WORD   T16LEN
     034604  015474                                                                .WORD   EXPREC
5728 034606                        120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034606  104406                                                                TRAP    C$CLP1
5729
5730 034610  004737  015774                JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
5731 034614  103405                        BCS     125$                    ;BR IF SOFT INIT OKAY
5732 034616  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5733 034620                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     034620  104455                                                                TRAP    C$ERDF
     034622  000772                                                                .WORD   506
     034624  003652                                                                .WORD   SFIERR
     034626  012034                                                                .WORD   SFIMSG
5734 034630                        125$:   CKLOOP                          ;LOOP IF SELECTED
     034630  104406                                                                TRAP    C$CLP1
5735 034632  000137  035016                JMP     300$                    ;
5736
5737                               ;    (* Verify Extended Features switch can be Inverted to CLEAR *)
5738 034636                        200$:
5739                               ;        Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
5740 034636  004737  037520                JSR     PC,T16SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
5741 034642  012704  037610                MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5742 034646  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5743 034652  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5744 034656                                FORCERROR       232$                    ;@@FORCE ERROR IF FORCER=1
5745 034672  103407                        BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
5746 034674  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5747 034676                                NEXT.ERRNO
5748 034676                        232$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     034676  104455                                                                TRAP    C$ERDF
     034700  000773                                                                .WORD   507
     034702  036317                                                                .WORD   T162SSR
     034704  012046                                                                .WORD   PKTSSR
5749 034706  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5750 034712                        240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034712  104406                                                                TRAP    C$CLP1
5751
```

```
5752                                    ;       DO a WRITE CHARACTERISTICS without an extended characteristic word
5753 034714  012704  037540                       MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5754 034720  012764  000016  000006               MOV     #14.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
5755 034726  004737  010662                       JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
5756 034732                                       FORCERROR       242$                    ;@@FORCE ERROR IF FORCER=1
5757 034746  103407                             BCS     250$                    ;BR IF CARRY SET (GOOD RETURN)
5758 034750  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5759 034752                                       NEXT.ERRNO
5760 034752                                242$:  ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     034752  104455                                                                    TRAP    C$ERDF
     034754  000774                                                                    .WORD   508
     034756  036262                                                                    .WORD   T16SSR
     034760  012046                                                                    .WORD   PKTSSR
5761 034762  005237  002222                       INC     FATFLG                  ;SET FATAL ERROR FLAG
5762 034766                                250$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034766  104406                                                                    TRAP    C$CLP1
5763                                    ;       If Message Buffer Data Length NOT= 10.  Then Print Error
5764 034770  013701  037564                       MOV     T16BFR+2,R1             ;GET RECV DATA FIELD LENGTH
5765 034774  012702  000012                       MOV     #10.,R2                 ;GET EXPD DATA FIELD LENGTH
5766 035000  020102                             CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
5767 035002  001404                             BEQ     270$                    ;ERROR IF NOT EQUAL
5768 035004                                       NEXT.ERRNO
5769 035004                                262$:  ERRHRD  ERRNO,T16LEN,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     035004  104456                                                                    TRAP    C$ERHRD
     035006  000775                                                                    .WORD   509
     035010  036532                                                                    .WORD   T16LEN
     035012  015474                                                                    .WORD   EXPREC
5770 035014                                270$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035014  104406                                                                    TRAP    C$CLP1
5771
5772
5773                                    ;       (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
5774                                    ;       Write to TSSR register to soft initialize the controller
5775 035016                                300$:
5776                                    ;       REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
5777 035016  012737  000001  002312       320$:  MOV     #1,DATA                 ;START AT BITS<21:19>=001
5778                                    ;       DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
5779 035024                                325$:
5780 035024  012704  037540                       MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5781 035030  012764  000016  000006               MOV     #14.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
5782 035036  013700  002312                       MOV     DATA,R0                 ;GET TEST DATA
5783         000003                               .REPT   3
5784                                             ASL     R0                      ;SHIFT INTO BITS 21:19
5785                                             .ENDR
5786 035050  010037  037552                       MOV     R0,T16DATA+2            ;STORE BUFFER ADDRESS BITS 21:19
5787 035054  010465  000000                       MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5788 035060  004737  016250                       JSR     PC,WAITF                ;WAIT FOR SSR
5789 035064                                       FORCERROR       342$                    ;@@FORCE ERROR IF FORCER=1
5790 035100  103407                             BCS     350$                    ;BR IF CARRY SET (GOOD RETURN)
5791 035102  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5792 035104                                       NEXT.ERRNO
5793 035104                                342$:  ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035104  104455                                                                    TRAP    C$ERDF
     035106  000776                                                                    .WORD   510
     035110  036262                                                                    .WORD   T16SSR
     035112  012046                                                                    .WORD   PKTSSR
5794 035114  005237  002222                       INC     FATFLG                  ;SET FATAL ERROR FLAG
```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```
5795 035120                          350$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035120  104406                                                                TRAP    C$CLP1
5796
5797                                 ;      If TSSR termination code NOT= Function Reject  Then Print Error
5798 035122  016501  000002                 MOV     TSSR(R5),R1             ;GET RECV TSSR
5799 035126  010102                         MOV     R1,R2                   ;COPY RECV TSSR
5800 035130  042702  000016                 BIC     #TERCLS,R2              ;CLEAR TC<2:0> EXPD
5801 035134  052702  000006                 BIS     #TSREJ,R2               ;SET EXPD TC<2:0>= FUNCTION REJECT
5802 035140                                 FORCERROR       352$,NOTSSR             ;@@FORCE ERROR IF FORCEP=1
5803 035150  020102                         CMP     R1,R2                   ;EXPD EQUAL RECV?
5804 035152  001404                         BEQ     360$                    ;BR IF YES
5805 035154                                 NEXT.ERRNO
5806 035154                          352$:  ERRHRD  ERRNO,T16REJ,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035154  104456                                                                TRAP    C$ERHRD
     035156  000777                                                                .WORD   511
     035160  036644                                                                .WORD   T16REJ
     035162  012046                                                                .WORD   PKTSSR
5807 035164                          360$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035164  104406                                                                TRAP    C$CLP1
5808 035166                                 FORCEXIT        370$
5809 035176  005237  002312                 INC     DATA                    ;GET NEXT TST PATTERN
5810 035202  023727  002312  000007         CMP     DATA,#7                 ;DONE ALL DATA?
5811 035210  101002                         BHI     370$                    ;BR IF YES
5812 035212  000137  035024                 JMP     325$                    ;DO ANOTHER TEST PATTERN
5813                                 ;      END-REPEAT
5814 035216                          370$:
5815 035216                                 ENDSUB                          ;////////////// END SUBTEST //////////
     035216                                                                         L10054:
     035216  104403                                                                TRAP    C$ESUB
5816
5817 035220  005737  002222                 TST     FATFLG                  ;ANY FATAL ERRORS ?
5818 035224  001402                         BEQ     460$                    ;BRANCH IF NOT
5819 035226  004737  017202                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
5820 035232                          460$:
5821
5822
5823
5824
5825
5826                                        .SBTTL  TEST  5: SUBTEST 2: VERIFY TIMERS A,B
5827
5828                                 ;++
5829                                 ; TEST 5: SUBTEST 2:
5830                                 ;
5831                                 ; SUBTEST DESCRIPTION:
5832                                 ;
5833                                 ;      This subtest verifies that timers A,B can be reset
5834                                 ;      and that Timer A is twice the frequency of Timer B.
5835                                 ;      Timer A has a period of 25 microseconds and Timer B
5836                                 ;      has a period of 50 microseconds. The timers are
5837                                 ;      checked at 1, 28, 53, and 78 micoseconds.
5838                                 ;
5839                                 ; TEST STEPS:
5840                                 ;
5841                                 ;
5842                                 ;      Write to TSSR register to soft initialize the controller
5843                                 ;      Do WRITE CHARACTERISTICS to setup a Message Buffer
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
                                        ;       (* Verify Timers A,B after RESET TIMER with 0 microsecond delay *)
                                        ;       Do a Write Control RESET TIMER with 1 microsecond delay
                                        ;       Do a Write Subsystem READ STATUS
                                        ;       If Timer A NOT= 0  Then Print Error
                                        ;       If Timer B NOT= 0  Then Print Error
                                        ;       (* Verify Timers A,B after RESET TIMER with 28 microsecond delay *)
                                        ;       Do a Write Control RESET TIMER with 28 microsecond delay
                                        ;       If Timer A NOT= 1  Then Print Error
                                        ;       If Timer B NOT= 1  Then Print Error
                                        ;       Do a Write Control RESET TIMER with 53 microsecond delay
                                        ;       If Timer A NOT= 0  Then Print Error
                                        ;       If Timer B NOT= 1  Then Print Error
                                        ;       Do a Write Control RESET TIMER with 78 microsecond delay
                                        ;       If Timer A NOT= 1  Then Print Error
                                        ;       If Timer B NOT= 0  Then Print Error
                                        ;--
5860 035232                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////
     035232                                                                                     T5.2:
     035232  104402                                                                                TRAP    C$BSUB
5861                                    ;       Write to TSSR register to soft initialize the controller
5862 035234                             5$:
5863 035234  004737  015774                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
5864 035240  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
5865 035242  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5866 035244                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     035244  104455                                                                             TRAP    C$ERDF
     035246  000777                                                                             .WORD   -11
     035250  003652                                                                             .WORD   SFIERR
     035252  012034                                                                             .WORD   SFIMSG
5867                                    ;       Do WRITE CHARACTERISTICS to setup a Message Buffer
5868 035254  004737  037360             10$:    JSR     PC,T16REST              ;RESTORE PACKET DEFAULTS
5869 035260  005037  002222                    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
5870 035264  012704  037540                    MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5871 035270  012764  000010  000006            MOV     #8.,PKBCNT(R4)          ;MESSAGE PACKET SIZE NO EXTEND
5872 035276  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
5873 035302                                    FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
5874 035316  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
5875 035320  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5876 035322                                    NEXT.ERRNO
5877 035322                             12$:   ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035322  104455                                                                             TRAP    C$ERDF
     035324  001000                                                                             .WORD   312
     035326  036262                                                                             .WORD   T16SSR
     035330  012046                                                                             .WORD   PKTSSR
5878 035332  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
5879 035336                             15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035336  104406                                                                             TRAP    C$CLP1
5880
5881                                    ;       (* Verify Timers A,B after RESET TIMER with 1 microsecond delay *)
5882                                    ;       Do a Write Control RESET TIMER with 1 microsecond delay
5883 035340  012700  000001                    MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
5884 035344  013701  036202                    MOV     T16001,R1               ;1 MICROSECOND DELAY
5885 035350  004737  037472                    JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
5886 035354  012704  037610                    MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5887 035360  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5888 035364  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5889 035370                                    FORCERROR       32$                     ;@@FORCE ERROR IF FORCER=1
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
5890 035404  103407                    BCS     40$                     ;BR IF CARRY SET (GOOD RETURN)
5891 035406  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5892 035410                            NEXT.ERRNO
5893 035410                    32$:    ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     035410  104455                                                            TRAP    C$ERDF
     035412  001001                                                            .WORD   513
     035414  036317                                                            .WORD   T162SSR
     035416  012046                                                            .WORD   PKTSSR
5894 035420  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
5895 035424                    40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035424  104406                                                            TRAP    C$CLP1
5896                           ;       If Timer A NOT= 0  Then Print Error
5897                           ;       If Timer B NOT= 0  Then Print Error
5898 035426  005002                    CLR     R2                      ;INIT EXPD
5899 035430  042702  000010            BIC     #S2.ATIM,R2             ;TIMER A EXPD=0
5900 035434  042702  000004            BIC     #S2.BTIM,R2             ;TIMER B EXPD=0
5901 035440  012700  037602            MOV     #T16BFSTA,R0            ;GET RECV READ STATUS
5902 035444  016001  000002            MOV     2(R0),R1                ;GET RECV BYTE 2
5903 035450  042701  177763            BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5904 035454                            FORCERROR       72$,NOTSSR      ;@@D
5905 035464  020201                    CMP     R2,R1                   ;EXPD EQUAL RECV?
5906 035466  001404                    BEQ     80$                     ;BR IF YES
5907 035470                            NEXT.ERRNO
5908 035470                    72$:    ERRHRD  ERRNO,T16T01,TIMEXP     ;REPORT ERROR
     035470  104456                                                            TRAP    C$ERHRD
     035472  001002                                                            .WORD   514
     035474  036761                                                            .WORD   T16T01
     035476  015552                                                            .WORD   TIMEXP
5909 035500                    80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035500  104406                                                            TRAP    C$CLP1
5910
5911                           ;       Do a Write Control RESET TIMER with 28 microsecond delay
5912 035502  012700  000001            MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
5913 035506  013701  036204            MOV     T16D28,R1               ;28 MICROSECOND DELAY
5914 035512  004737  037472            JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
5915 035516  012704  037610            MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5916 035522  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5917 035526  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5918 035532                            FORCERROR       112$            ;@@DFORCE ERROR IF FORCER=1
5919 035546  103407                    BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
5920 035550  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5921 035552                            NEXT.ERRNO
5922 035552                    112$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     035552  104455                                                            TRAP    C$ERDF
     035554  001003                                                            .WORD   515
     035556  036317                                                            .WORD   T162SSR
     035560  012046                                                            .WORD   PKTSSR
5923 035562  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
5924 035566                    120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035566  104406                                                            TRAP    C$CLP1
5925                           ;       If Timer A NOT= 1  Then Print Error
5926                           ;       If Timer B NOT= 1  Then Print Error
5927 035570  005002                    CLR     R2                      ;INIT EXPD
5928 035572  052702  000010            BIS     #S2.ATIM,R2             ;TIMER A EXPD=1
5929 035576  052702  000004            BIS     #S2.BTIM,R2             ;TIMER B EXPD=1
5930 035602  012700  037602            MOV     #T16BFSTA,R0            ;GET RECV READ STATUS
5931 035606  016001  000002            MOV     2(R0),R1                ;GET RECV BYTE 2
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
5932 035612  042701  177763                BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5933 035616                                FORCERROR       172$,NOTSSR     ;@@D
5934 035626  020201                        CMP     R2,R1                   ;EXPD EQUAL RECV?
5935 035630  001404                        BEQ     180$                    ;BR IF YES
5936 035632                                NEXT.ERRNO
5937 035632                          172$: ERRHRD  ERRNO,T16T28,TIMEXP     ;REPORT ERROR
     035632  104456                                                              TRAP    C$ERHRD
     035634  001004                                                              .WORD   516
     035636  037060                                                              .WORD   T16T28
     035640  015552                                                              .WORD   TIMEXP
5938 035642                          180$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035642  104406                                                              TRAP    C$CLP1
5939
5940                                 ;     Do a Write Control RESET TIMER with 53 microsecond delay
5941 035644  012700  000001                MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
5942 035650  013701  036206                MOV     T16D53,R1               ;53 MICROSECOND DELAY
5943 035654  004737  037472                JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
5944 035660  012704  037610                MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5945 035664  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5946 035670  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5947 035674                                FORCERROR       212$            ;@@DFORCE ERROR IF FORCER=1
5948 035710  103407                        BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
5949 035712  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5950 035714                                NEXT.ERRNO
5951 035714                          212$: ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     035714  104455                                                              TRAP    C$ERDF
     035716  001005                                                              .WORD   517
     035720  036317                                                              .WORD   T162SSR
     035722  012046                                                              .WORD   PKTSSR
5952 035724  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5953 035730                          220$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035730  104406                                                              TRAP    C$CLP1
5954                                 ;     If Timer A NOT= 0  Then Print Error
5955                                 ;     If Timer B NOT= 1  Then Print Error
5956 035732  005002                        CLR     R2                      ;INIT EXPD
5957 035734  042702  000010                BIC     #S2.ATIM,R2             ;TIMER A EXPD=0
5958 035740  052702  000004                BIS     #S2.BTIM,R2             ;TIMER B EXPD=1
5959 035744  012700  037602                MOV     #T16BFSTA,R0            ;GET RECV READ STATUS
5960 035750  016001  000002                MOV     2(R0),R1                ;GET RECV BYTE 2
5961 035754  042701  177763                BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5962 035760                                FORCERROR       272$,NOTSSR     ;@@D
5963 035770  020201                        CMP     R2,R1                   ;EXPD EQUAL RECV?
5964 035772  001404                        BEQ     280$                    ;BR IF YES
5965 035774                                NEXT.ERRNO
5966 035774                          272$: ERRHRD  ERRNO,T16T53,TIMEXP     ;REPORT ERROR
     035774  104456                                                              TRAP    L$ERHRD
     035776  001006                                                              .WORD   518
     036000  037160                                                              .WORD   T16T53
     036002  015552                                                              .WORD   TIMEXP
5967 036004                          280$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036004  104406                                                              TRAP    C$CLP1
5968                                 ;     Do a Write Control RESET TIMER with 78 microsecond delay
5969 036006  012700  000001                MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
5970 036012  013701  036210                MOV     T16D78,R1               ;78 MICROSECOND DELAY
5971 036016  004737  037472                JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
5972 036022  012704  037610                MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5973 036026  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
5974 036032  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5975 036036                                FORCERROR       312$                    ;@@FORCE ERROR IF FORCER=1
5976 036052  103407                        BCS     320$                    ;BR IF CARRY SET (GOOD RETURN)
5977 036054  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5978 036056                                NEXT.ERRNO
5979 036056                        312$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036056  104455                                                                TRAP    C$ERDF
     036060  001007                                                                .WORD   519
     036062  036317                                                                .WORD   T162SSR
     036064  012046                                                                .WORD   PKTSSR
5980 036066  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5981 036072                        320$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036072  104406                                                                TRAP    C$CLP1
5982                                ;       If Timer A NOT= 1  Then Print Error
5983                                ;       If Timer B NOT= 0  Then Print Error
5984 036074  005002                        CLR     R2                      ;INIT EXPD
5985 036076  052702  000010                BIS     #S2.ATIM,R2             ;TIMER A EXPD=1
5986 036102  042702  000004                BIC     #S2.BTIM,R2             ;TIMER B EXPD=0
5987 036106  012700  037602                MOV     #T168FSTA,R0            ;GET RECV READ STATUS
5988 036112  016001  000002                MOV     2(R0),R1                ;GET RECV BYTE 2
5989 036116  042701  177763                BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5990 036122                                FORCERROR       372$,NOTSSR     ;@@D
5991 036132  020201                        CMP     R2,R1                   ;EXPD EQUAL RECV?
5992 036134  001404                        BEQ     380$                    ;BR IF YES
5993 036136                                NEXT.ERRNO
5994 036136                        372$:   ERRHRD  ERRNO,T16T78,TIMEXP     ;REPORT ERROR
     036136  104456                                                                TRAP    C$ERHRD
     036140  001010                                                                .WORD   520
     036142  037260                                                                .WORD   T16T78
     036144  015552                                                                .WORD   TIMEXP
5995 036146                        380$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036146  104406                                                                TRAP    C$CLP1
5996
5997 036150                                ENDSUB                          ;////////////// END SUBTEST //////////
     036150                                                                  L10055:
     036150  104403                                                                TRAP    C$ESUB
5998
5999 036152  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
6000 036156  001402                        BEQ     460$                    ;BRANCH IF NOT
6001 036160  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
6002 036164  004737  016456        460$:   JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS?
6003 036170  103002                        BCC     465$                    ;BR IF NO
6004 036172  000137  034152                JMP     T16LOOP                 ;LOOP UNTIL ITERATIONS DONE
6005 036176                        465$:
6006
6007
6008 036176                                EXIT    TST                     ;////////// EXIT TEST //////////
     036176  104432                                                                TRAP    C$EXIT
     036200  001524                                                                .WORD   L10053-.
6009
6010
6011                                ;+
6012                                ;LOCAL STORAGE FOR THIS TEST
6013                                ;-
6014 036202  000001                T16D01:         .WORD   1               ;1 MICROSECOND DELAY (ACTUALLY .8 MIC)
6015 036204  000040                T16D28:         .WORD   40              ;28 MICROSECOND DELAY (.8 MICROS PER)
6016 036206  000076                T16D53:         .WORD   76              ;53 MICROSECOND
```

```
 6017 036210  000142                    T16D78:         .WORD   142             ;78 MICROSECOND
 6018                                    ;+
 6019                                    ;LOCAL TEXT MESSAGES FOR TEST
 6020                                    ;-
 6021
 6022 036212      105     170     164    TST16ID:        .ASCIZ  'Extended Features Switch and Timers A,B'
 6023 036262      127     122     111    T16SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
 6024 036317      127     122     111    T162SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
 6025 036363      127     122     111    T163SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
 6026 036430      102     165     163    T16TSBA:.ASCIZ  'Bus Address Register (TSBA) Incorrect after Write Characteristics'
 6027 036532      104     141     164    T16LEN:.ASCIZ   'Data Field Length in Message Buffer Incorrect after Write Characteristics'
 6028 036644      124     123     123    T16REJ:.ASCIZ   'TSSR Function Reject Not Returned When Non-Existent Buffer Address Specifie
d'
 6029 036761      124     151     155    T16T01:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 1 microsecond Delay'
 6030 037060      124     151     155    T16T28:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 28 microsecond Delay'
 6031 037160      124     151     155    T16T53:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 53 microsecond Delay'
 6032 037260      124     151     155    T16T78:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 78 microsecond Delay'
 6033                                            .EVEN
 6034
 6035                                    ;+
 6036                                    ; SET DEFAULT PACKET
 6037                                    ;-
 6038 037360                            T16REST:
 6039 037360  012700  037540                    MOV     #T16PACKET,R0           ;PACKET ADDRESS
 6040 037364  012720  100004                    MOV     #100004,(R0)+           ;WRITE CHARACTERISTICS WITH ACK
 6041 037370  012720  037550                    MOV     #T16DATA,(R0)+          ;ADDRESS OF CHAR DATA BLOCK
 6042 037374  005020                            CLR     (R0)+                   ;EXTENDED ADDRESS
 6043 037376  012720  000012                    MOV     #10.,(R0)+              ;SIZE OF MESSAGE PACKET
 6044 037402  012720  037562                    MOV     #T168FR,(R0)+           ;MESSAGE BUFFER ADDRESS
 6045 037406  005020                            CLR     (R0)+                   ;CLEAR EXTENDED BUFFER ADDRESS
 6046 037410  012720  000024                    MOV     #20.,(R0)+              ;LENGTH OF MESSAGE BUFFER
 6047 037414  005020                            CLR     (R0)+                   ;CLEAR ESS,ENB,EAI,ERI
 6048 037416  005010                            CLR     (R0)                    ;CLEAR EXTENDED FEATURES WORD
 6049 037420  005037  037562                    CLR     T168FR                  ;CLEAR 1ST LOCATION IN MESSAGE BUFFER
 6050 037424  000207                            RTS     PC                      ;
 6051
 6052                                    ;+
 6053                                    ; CLEAR MESSAGE BUFFER
 6054                                    ;-
 6055 037426                            T16CLRBUF:                              ;
 6056 037426                                    SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
 6057 037432  012701  037562                    MOV     #T168FR,R1              ;GET MESSAGE BUFFER ADDRESS
 6058 037436  012702  000026                    MOV     #T168END-T16BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES
 6059 037442  105021                    10$:    CLRB    (R1)+                   ;CLEAR A BYTE
 6060 037444  005302                            DEC     R2                      ;DONE?
 6061 037446  003375                            BGT     10$                     ;BR IF NO
 6062 037450  000207                            RTS     PC                      ;RETURN
 6063
 6064                                    ;+
 6065                                    ; SETUP T16PK2 PACKET FOR READ STATUS
 6066                                    ;-
 6067 037452                            T16SRD:
 6068 037452  004737  037426                    JSR     PC,T16CLRBUF            ;CLEAR MESSAGE BUFFER
 6069 037456  012700  037620                    MOV     #T16D^2,R0              ;WRITE SUBSYSTEM DATA BUFFER
 6070 037462  112720  000005                    MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
 6071 037466  105010                            CLRB    (R0)                    ;CLEAR BSEL1
 6072 037470  000207                            RTS     PC                      ;RETURN
 6073
```

```
6074
6075                                    ;*
6076                                    ; SETUP T16PK2 PACKET FOR WRITE MISC.
6077                                    ;
6078                                    ; INPUT:
6079                                    ;       R0      CONTAINS WRITE MISC FUNCTION CODE (BSEL1)
6080                                    ;       R1      CONTAINS DELAY (TIMES 800 NS) FOR BSEL2
6081                                    ;-
6082 037472                             T16WMISC:
6083 037472                                     SAVREG                          ;SAVE R1-R5 UNITL NEXT RETURN
6084 037476  004737  037426                     JSR     PC,T16CLRBUF            ;CLEAR MESSAGE BUFFER
6085 037502  012702  037620                     MOV     #T16DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
6086 037506  112722  000010                     MOVB    #PW.WMISC,(R2)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
6087 037512  110022                             MOVB    R0,(R2)+                ;STORE WRITE MISC CODE IN BSEL1
6088 037514  110112                             MOVB    R1,(R2)                 ;STORE DELAY (RESET TIMER) IN BSEL2
6089 037516  000207                             RTS     PC                      ;RETURN
6090                                    ;*
6091                                    ; SETUP T16PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
6092                                    ;-
6093 037520                             T16SEXT:
6094 037520  012700  037620                     MOV     #T16DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
6095 037524  112720  000010                     MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
6096 037530  112710  000200                     MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
6097 037534  000207                             RTS     PC                      ;RETURN
6098
6099
6100
6101
6103         037540                             .=<.+10>&177770
6105                                    ;
6106                                    ;WRITE CHARACTERISTICS COMMAND PACKET
6107                                    ;
6108 037540                             T16PACKET:                              ;COMMAND PACKET FOR TEST
6109 037540  100004                             .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
6110 037542  037550                             .WORD   T16DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
6111 037544  000000                             .WORD   0
6112 037546  000012                             .WORD   10.                     ;MESSAGE PACKET ' 2E
6113
6114 037550                             T16DATA:                                ;CHARACTERISTICS DATA BLOCK
6115 037550  037562                             .WORD   T16BFR                  ;ADDRESS OF MESSAGE BUFFER
6116 037552  000000                             .WORD   0
6117 037554  000024                             .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
6118 037556  000000                             .WORD   0                       ;ESS,ENB,EAI,ERI
6119 037560  000000                             .WORD   0                       ;EXTENDED FEATURES WORD
6120
6121
6122                                    ;MESSAGE BUFFER
6123
6124 037562                             T16BFR:                                 ;BEGIN MESSAGE BUFFER
6125 037562  000000                             .WORD   0                       ;MESSAGE TYPE
6126 037564  000000                             .WORD   0                       ;DATA FIELD LENGTH
6127 037566  000000                             .WORD   0                       ;RBPCR
6128 037570  000000                             .WORD   0                       ;XST0
6129 037572  000000                             .WORD   0                       ;XST1
6130 037574  000000                             .WORD   0                       ;XST2
6131 037576  000000                             .WORD   0                       ;XST3
6132 037600  000000                             .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
   6133 037602                          T16BFSTA: .BLKB 6.                      ;READ STATUS AND WRITE FIFO BUFFER
   6134 037610                          T16BEND:                                ;END OF MESSAGE BUFFER
   6135                                 ;
   6136                                 ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
   6137                                 ;
   6141 037610                          T16PK2:
   6142 037610  100006                          .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
   6143 037612  037620                          .WORD   T16DT2                  ;LOW ADDRESS OF DATA BLOCK
   6144 037614  000000                          .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
   6145 037616  000012                          .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
   6146
   6147 037620                          T16DT2:                                 ;DATA BLOCK
   6148 037620     000                          .BYTE   0                       ;BSEL0
   6149 037621     000                          .BYTE   0                       ;BSEL1
   6150 037622  000000                          .WORD   0                       ;SEL2
   6151 037624                                  .BLKB   64.                     ;WRITE FIFO DATA OUTPUT BUFFER
   6152
   6153
   6154 037724                          ENDTST
        037724                                                          L10053:
        037724  104401                                                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST  6: 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
                                ;--

 037726                                 BGNTST
 037726                                                                  T6::
 037726  012700  046156                 MOV     @TST17ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
 037732  004737  016510                 JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
 037736  012737  000012  002216         MOV     @10.,LOOPCNT            ;PERFORM 10 ITERATIONS
 037744  004737  017274                 JSR     PC,KTOFF                ;SHUT OFF MEMORY MANAGEMENT
 037750  005037  003134                 CLR     KTENABLE                ;REALLY SHUT DOWN KT-11
 037754                         T17LOOP:



                                        .SBTTL  TEST  6: SUBTEST 1: FIFO INITIALIZE STATUS TEST
                                ;**
                                ; TEST 6: 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
```

TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

```
6216                                    ;--
6217 037754                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////
     037754                                                                             T6.1:
     037754  104402                                                                        TRAP    C$BSUB
6218
6219                                    ;       Write to TSSR register to soft initialize the controller
6220 037756                             5$:
6221 037756  004737  015774                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
6222 037762  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
6223 037764  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6224 037766                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     037766  104455                                                                         TRAP    C$ERDF
     037770  001130                                                                        .WORD   600
     037772  003652                                                                        .WORD   SFIERR
     037774  012034                                                                        .WORD   SFIMSG
6225                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
6226 037776  005037  002222             10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
6227 040002  012704  047550                    MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
6228 040006  C04737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
6229 040012                                    FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
6230 040026  103407                            BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
6231 040030  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6232 040032                                    NEXT.ERRNO
6233 040032                             42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     040032  104455                                                                         TRAP    C$ERDF
     040034  001131                                                                        .WORD   601
     040036  046175                                                                        .WORD   T17SSR
     040040  012046                                                                        .WORD   PKTSSR
6234 040042  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
6235 040046                             50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040046  104406                                                                         TRAP    C$CLP1
6236
6237                                    ;       Do a Write Subsystem READ STATUS
6238 040050  004737  047334                    JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6239 040054  012704  047720                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6240 040060  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6241 040064  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6242 040070                                    FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
6243 040104  103407                            BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
6244 040106  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6245 040110                                    NEXT.ERRNO
6246 040110                             62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     040110  104455                                                                         TRAP    C$ERDF
     040112  001132                                                                        .WORD   602
     040114  046276                                                                        .WORD   T173SSR
     040116  012046                                                                        .WORD   PKTSSR
6247 040120  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
6248 040124                             70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040124  104406                                                                         TRAP    C$CLP1
6249                                    ;       Set WORDS 0-7  of expd message buffer = to recv since not testing
6250 040126  004737  047516                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
6251 040132  012701  045752                    MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
6252 040136  012702  047612                    MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
6253 040142  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
6254 040144  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
6255 040146  052711  000020                    BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
6256 040152  042711  000040                    BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
```

TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

```
6257 040156  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
6258                                ;       If Input Ready NOT=1 then Print Error
6259                                ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6260 040162  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6261 040164  012701  047572                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
6262 040170  012702  045732                MOV     #T17EXP,R2              ;EXPD ADDRESS
6263 040174  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
6264 040200  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6265 040204                                FORCERROR       82$,NOTSSR      ;SSD
6266 040214  103404                        BCS     90$                     ;BR IF YES
6267 040216                                NEXT.ERRNO
6268 040216                        82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
     040216  104456                                                                TRAP    C$ERHRD
     040220  001133                                                                .WORD   603
     040222  046515                                                                .WORD   T171CMP
     040224  012350                                                                .WORD   MSGSTAT
6269 040226                        90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040226  104406                                                                TRAP    C$CLP1
6270
6271 040230                                ENDSUB                          ;////////////// END SUBTEST //////////
     040230                                                                  L10057:
     040230  104403                                                                TRAP    C$ESUB
6272
6273 040232  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
6274 040236  001402                        BEQ     160$                    ;BRANCH IF NOT
6275 040240  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
6276 040244                        160$:
6277
6278                                        .SBTTL  TEST  6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST
6279
6280                                ;++
6281                                ; TEST 6: SUBTEST 2:
6282                                ;
6283                                ; SUBTEST DESCRIPTION:
6284                                ;
6285                                ;       This subtest verifies the ability of the FIFO to correctly
6286                                ;       pass a single data byte from input to output. For each
6287                                ;       of 256 data values (0-377 octal) the following is done:
6288                                ;               1. Initial FIFO status is checked
6289                                ;               2. The Write FIFO function, specifying a count of
6290                                ;                  one byte to be written is executed.
6291                                ;               3. Read Status is executed and FIFO status is checked.
6292                                ;               4. Read FIFO is executed and the data and final status
6293                                ;                  is checked.
6294                                ;
6295                                ; TEST STEPS:
6296                                ;
6297                                ; BEGIN
6298                                ;       Write to TSSR to soft initialize
6299                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
6300                                ;       Do a Write Subsystem READ STATUS
6301                                ;       If Input Ready  NOT=1   Then Print Error
6302                                ;       If Output Ready NOT=0   Then Print Error
6303                                ;       If Data In Miss NOT=0   Then Print Error
6304                                ;
6305                                ;  REPEAT FOR DATA FROM 0 TO 377 OCTAL
6306                                ;    BEGIN
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
6307                                    ;       Do a Write Subsystem WRITE NPR to set tape direction out
6308                                    ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
6309                                    ;       Do a Write Subsystem READ STATUS
6310                                    ;       If Input Ready  NOT=1   Then Print Error
6311                                    ;       If Output Ready NOT=1   Then Print Error
6312                                    ;       If Data In Miss NOT=0   Then Print Error
6313                                    ;       Do Write Subsystem READ FIFO with byte count equal to 1
6314                                    ;       If Data read from FIFO NOT= to Data sent Then Print Error
6315                                    ;       Do a Write Subsystem READ STATUS
6316                                    ;       If Input Ready  NOT=1   Then Print Error
6317                                    ;       If Output Ready NOT=0   Then Print Error
6318                                    ;       If Data In Miss NOT=0   Then Print Error
6319                                    ;   END
6320                                    ; END
6321                                    ;--
6322 040244                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      040244                                                                                    T6.2:
      040244  104402                                                                            TRAP    C$BSUB
6323
6324                                    ;       Write to TSSR register to soft initialize the controller
6325 040246                            5$:
6326 040246  004737  015774                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
6327 040252  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
6328 040254  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6329 040256                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      040256  104455                                                                            TRAP    C$ERDF
      040260  001133                                                                            .WORD   603
      040262  003652                                                                            .WORD   SFIERR
      040264  012034                                                                            .WORD   SFIMSG
6330                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
6331 040266  005037  002222            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
6332 040272  012704  047550                    MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
6333 040276  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
6334 040302                                    FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
6335 040316  103407                            BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 336 040320  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 337 040322                                    NEXT.ERRNO
6338 040322                            42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      040322  104455                                                                            TRAP    C$ERDF
      040324  001134                                                                            .WORD   604
      040326  046175                                                                            .WORD   T17SSR
      040330  012046                                                                            .WORD   PKTSSR
6339 040332  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
6340 040336                            50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      040336  104406                                                                            TRAP    C$CLP1
6341                                    ;       Do a Write Subsystem READ STATUS
6342 040340  004737  047334                    JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6343 040344  012704  047720                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6344 040350  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6345 040354  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6346 040360                                    FORCERROR       62$                     ;@@FORCE ERROR IF FORCER=1
6347 040374  103407                            BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
6348 040376  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6349 040400                                    NEXT.ERRNO
6350 040400                            62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      040400  104455                                                                            TRAP    C$ERDF
      040402  001135                                                                            .WORD   605
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
        040404  046276                                                          .WORD   T173SSR
        040406  012046                                                          .WORD   PKTSSR
 6351   040410  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
 6352   040414                      70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        040414  104406                                                          TRAP    C$CLP1
 6353                               ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
 6354   040416  004737  047516              JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
 6355   040422  012701  045752              MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 6356   040426  012702  047612              MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 6357   040432  012221                      MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 6358   040434  011211                      MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 6359   040436  052711  000020              BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
 6360   040442  042711  000040              BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
 6361   040446  042711  000200              BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
 6362                               ;       If Input Ready NOT=1 then Print Error
 6363                               ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
 6364   040452  005000                      CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 6365   040454  012701  047572              MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 6366   040460  012702  045732              MOV     #T17EXP,R2              ;EXPD ADDRESS
 6367   040464  012703  000024              MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 6368   040470  004737  011500              JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 6369   040474                              FORCERROR       82$,NOTSSR      ;@@D
 6370   040504  103404                      BCS     90$                     ;BR IF YES
 6371   040506                              NEXT.ERRNO
 6372   040506                      82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
        040506  104456                                                          TRAP    C$ERHRD
        040510  001136                                                          .WORD   606
        040512  046515                                                          .WORD   T171CMP
        040514  012350                                                          .WORD   MSGSTAT
 6373   040516                      90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        040516  104406                                                          TRAP    C$CLP1
 6374
 6375                               ;  Repeat for DATA from 0 to 377
 6376   040520  012737  000000  002312          MOV     #0,DATA             ;GET FIRST DATA
 6377   040526                      100$:                                   ;REPEAT LABEL
 6378                               ;       Do a Write Subsystem WRITE NPR to set tape direction out
 6379   040526  012700  000100              MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
 6380   040532  004737  047376              JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
 6381   040536  012704  047720              MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 6382   040542  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 6383   040546  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6384   040552                              FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
 6385   040566  103407                      BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
 6386   040570  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6387   040572                              NEXT.ERRNO
 6388   040572                      102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        040572  104455                                                          TRAP    C$ERDF
        040574  001137                                                          .WORD   607
        040576  046343                                                          .WORD   T174SSR
        040600  012046                                                          .WORD   PKTSSR
 6389   040602  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
 6390   040606                      105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        040606  104406                                                          TRAP    C$CLP1
 6391                               ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
 6392   040610  012700  000001              MOV     #1,R0                   ;WRITE 1 BYTE
 6393   040614  012701  002312              MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
 6394   040620  004737  047422              JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
 6395 040624  012704  047720                    MOV     @T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 6396 040630  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 6397 040634  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6398 040640                                    FORCERROR       107$                    ;@@@FORCE ERROR IF FORCER=1
 6399 040654  103407                            BCS     11C$                    ;BR IF CARRY SET (GOOD RETURN)
 6400 040656  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6401 040660                                    NEXT.ERRNO
 6402 040660                            107$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      040660  104455                                                                  TRAP    C$ERDF
      040662  001140                                                                  .WORD   608
      040664  046406                                                                  .WORD   T175SSR
      040666  012046                                                                  .WORD   PKTSSR
 6403 040670  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
 6404 040674                            110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      040674  104406                                                                  TRAP    C$CLP1
 6405
 6406                                   ;       Do a Write Subsystem READ STATUS
 6407 040676  004737  047334                    JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
 6408 040702  C12704  047720                    MOV     @T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 6409 040706  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 6410 040712  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6411 040716                                    FORCERROR       112$                    ;@@@FORCE ERROR IF FORCER=1
 6412 040732  103407                            BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
 6413 040734  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6414 040736                                    NEXT.ERRNO
 6415 040736                            112$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      040736  104455                                                                  TRAP    C$ERDF
      040740  001141                                                                  .WORD   609
      040742  046276                                                                  .WORD   T173SSR
      040744  012046                                                                  .WORD   PKTSSR
 6416 040746  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
 6417 040752                            120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      040752  104406                                                                  TRAP    C$CLP1
 6418                                   ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
 6419 040754  004737  047516                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
 6420 040760  012701  045752                    MOV     @T17EXSTA,R1            ;GET EXPECTED READ STATUS
 6421 040764  012702  047612                    MOV     @T17BFSTA,R2            ;GET RECV READ STATUS
 6422 040770  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD @8 = RECV TEMP
 6423 040772  011211                            MOV     (R2),(R1)               ;SET EXPD WORD @9 = RECV TEMP
 6424 040774  052711  000020                    BIS     @S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
 6425 041000  052711  000040                    BIS     @S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
 6426 041004  042711  000200                    BIC     @S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
 6427                                   ;       If Input Ready NOT=1 then Print Error
 6428                                   ;       If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
 6429 041010  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 6430 041012  012701  047572                    MOV     @T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 6431 041016  012702  045732                    MOV     @T17EXP,R2              ;EXPD ADDRESS
 6432 041022  012703  000024                    MOV     @20.,R3                 ;NUMBER OF BYTES TO COMPARE
 6433 041026  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 6434 041032                                    FORCERROR       132$,NOTSSR             ;@@@
 6435 041042  103404                            BCS     140$                    ;BR IF YES
 6436 041044                                    NEXT.ERRNO
 6437 041044                            132$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
      041044  104456                                                                  TRAP    C$ERHRD
      041046  001142                                                                  .WORD   610
      041050  046673                                                                  .WORD   T173CMP
      041052  012350                                                                  .WORD   MSGSTAT
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
6438 041054                          140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041054  104406                                                                  TRAP    C$CLP1
6439
6440                                 ;       Do Write Subsystem READ FIFO with byte count equal to 1
6441 041056  012700  000001                  MOV     #1,R0                   ;SET READ BYTE COUNT
6442 041062  004737  047456                  JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
6443 041066  012704  047720                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6444 041072  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6445 041076  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6446 041102                                  FORCERROR       142$                    ;@@FORCE ERROR IF FORCER=1
6447 041116  103407                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
6448 041120  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6449 041122                                  NEXT.ERRNO
6450 041122                          142$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041122  104455                                                                  TRAP    C$ERDF
     041124  001143                                                                  .WORD   611
     041126  046452                                                                  .WORD   T176SSR
     041130  012046                                                                  .WORD   PKTSSR
6451 041132  C05237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
6452 041136                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041136  104406                                                                  TRAP    C$CLP1
6453                                 ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
6454 041140  004737  047516                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
6455 041144  012701  045752                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
6456 041150  012702  047612                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
6457 041154  013721  002312                  MOV     DATA,(R1)+              ;SET EXPD WORD #8 = COUNT DATA
6458 041160  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV (NOT TESTING)
6459                                 ;       If Data read from FIFO NOT= to Data sent Then Print Error
6460                                 ;       The data is in WORD #8 of the message buffer
6461 041162  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6462 041164  012701  047572                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
6463 041170  012702  045732                  MOV     #T17EXP,R2              ;EXPD ADDRESS
6464 041174  012703  000022                  MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
6465 041200  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6466 041204                                  FORCERROR       152$,NOTSSR             ;@@D
6467 041214  103404                          BCS     160$                    ;BR IF YES
6468 041216                                  NEXT.ERRNO
6469 041216                          152$:   ERRHRD  ERRNO,T172CMP,MSGSUB    ;REPORT ERROR
     041216  104456                                                                  TRAP    C$ERHRD
     041220  001144                                                                  .WORD   612
     041222  046577                                                                  .WORD   T172CMP
     041224  013742                                                                  .WORD   MSGSUB
6470 041226                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041226  104406                                                                  TRAP    C$CLP1
6471
6472                                 ;       Do a Write Subsystem READ STATUS
6473 041230  004737  047334                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6474 041234  012704  047720                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6475 041240  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6476 041244  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6477 041250                                  FORCERROR       162$                    ;@@FORCE ERROR IF FORCER=1
6478 041264  103407                          BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
6479 041266  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6480 041270                                  NEXT.ERRNO
6481 041270                          162$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041270  104455                                                                  TRAP    C$ERDF
     041272  001145                                                                  .WORD   613
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
        041274  046276                                                          .WORD   T173SSR
        041276  012046                                                          .WORD   PKTSSR
  6482  041300  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
  6483  041304                    170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        041304  104406                                                          TRAP    C$CLP1
  6484                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
  6485  041306  004737  047516            JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
  6486  041312  012701  045752            MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
  6487  041316  012702  047612            MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
  6488  041322  012221                    MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
  6489  041324  011211                    MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
  6490  041326  052711  000020            BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
  6491  041332  042711  000040            BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
  6492  041336  042711  000200            BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
  6493                            ;       If Input Ready NOT=1 then Print Error
  6494                            ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
  6495  041342  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
  6496  041344  012701  047572            MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
  6497  041350  012702  045732            MOV     #T17EXP,R2              ;EXPD ADDRESS
  6498  041354  012703  000024            MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
  6499  041360  004737  011500            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
  6500  041364                            FORCERROR       172$,NOTSSR     ;@@D
  6501  041374  103404                    BCS     180$                    ;BR IF YES
  6502  041376                            NEXT.ERRNO
  6503  041376                    172$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
        041376  104456                                                          TRAP    C$ERHRD
        041400  001146                                                          .WORD   614
        041402  046757                                                          .WORD   T174CMP
        041404  012350                                                          .WORD   MSGSTAT
  6504  041406                    180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        041406  104406                                                          TRAP    C$CLP1
  6505  041410                            FORCEXIT        205$            ;@@D
  6506  041420  005237  002312            INC     DATA                    ;GET NEXT TEST DATA
  6507  041424  023727  002312  000377      CMP     DATA,#377               ;DONE 0 TO 377?
  6508  041432  101002                    BHI     205$                    ;BR IF YES
  6509  041434  000137  040526            JMP     100$                    ;DO ANOTHER TEST PATTERN
  6510  041440                    205$:
  6511
  6512  041440                            ENDSUB                          ;////////////// END SUBTEST //////////
        041440                                                                  L10060:
        041440  104403                                                          TRAP    C$ESUB
  6513
  6514  041442  005737  002222            TST     FATFLG                  ;ANY FATAL ERRORS ?
  6515  041446  001402                    BEQ     260$                    ;BRANCH IF NOT
  6516  041450  004737  017202            JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
  6517  041454                    260$:
  6518
  6519                            .SBTTL  TEST  6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST
  6520
  6521                            ;++
  6522                            ; TEST 6: SUBTEST 3:
  6523                            ;
  6524                            ; SUBTEST DESCRIPTION:
  6525                            ;
  6526                            ;       This subtest verifies the ability of the FIFO to correctly
  6527                            ;       pass a multiple data bytes from input to output.
  6528                            ;       The following sequence is done with various data patterns
```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
                                ;       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
                                ;--
041454                          ;       BGNSUB                          ;////////////// BEGIN SUBTEST //////////
041454                                                                                  T6.3:
041454  104402                                                                                  TRAP    C$BSUB
                                ;
                                ;       Write to TSSR register to soft initialize the controller
                                5$:
041456  004737  015774          ;       JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
041462  103405                  ;       BCS     10$                     ;BR IF SOFT INIT OKAY
041464  010001                  ;       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041466                          ;       ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
041466  104455                                                                          TRAP    C$ERDF
041470  001146                                                                          .WORD   614
041472  003652                                                                          .WORD   SFIERR
041474  012034                                                                          .WORD   SFIMSG
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
041476  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
041502  012704  047550          ;       MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
041506  004737  010662          ;       JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
041512                          ;       FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
041526  103407                  ;       BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
6580 041530  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6581 041532                            NEXT.ERRNO
6582 041532                    42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     041532  104455                                                            TRAP    C$ERDF
     041534  001147                                                            .WORD   615
     041536  046175                                                            .WORD   T17SSR
     041540  012046                                                            .WORD   PKTSSR
6583 041542  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
6584 041546                    50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041546  104406                                                            TRAP    C$CLP1
6585                           ;       Do a Write Subsystem READ STATUS
6586 041550  004737  047334            JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6587 041554  012704  047720            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6588 041560  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6589 041564  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6590 041570                            FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
6591 041604  103407                    BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
6592 041606  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6593 041610                            NEXT.ERRNO
6594 041610                    62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041610  104455                                                            TRAP    C$ERDF
     041612  001150                                                            .WORD   616
     041614  046276                                                            .WORD   T173SSR
     041616  012046                                                            .WORD   PKTSSR
6595 041620  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
6596 041624                    70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041624  104406                                                            TRAP    C$CLP1
6597                           ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
6598 041626  004737  047516            JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
6599 041632  012701  045752            MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
6600 041636  012702  047612            MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
6601 041642  012221                    MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
6602 041644  011211                    MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
6603 041646  052711  000020            BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY=  1
6604 041652  042711  000040            BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY=  0
6605 041656  042711  000200            BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
6606 041662  042711  000100            BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
6607                           ;       If Input Ready NOT=1 then Print Error
6608                           ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6609                           ;       If Last Word    NOT=0   Then Print Error
6610 041666  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6611 041670  012701  047572            MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
6612 041674  012702  045732            MOV     #T17EXP,R2              ;EXPD ADDRESS
6613 041700  012703  000024            MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
6614 041704  004737  011500            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6615 041710                            FORCERROR       82$,NOTSSR      ;@@D
6616 041720  103404                    BCS     90$                     ;BR IF YES
6617 041722                            NEXT.ERRNO
6618 041722                    82$:    ERRMRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
     041722  104456                                                            TRAP    C$ERMRD
     041724  001151                                                            .WORD   617
     041726  046515                                                            .WORD   T171CMP
     041730  012350                                                            .WORD   MSGSTAT
6619 041732                    90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041732  104406                                                            TRAP    C$CLP1
6620
6621
```

```
                                        ;  REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
                                        ;  TSTFLAG =1 FOR INCREMENT TEST PATTERN
                                        ;           =2 FOR DECREMENT TEST PATTERN
                                        ;           =3 FOR TSTBLK TABLE PATTERN
041734  012737  000001  002314          95$:    MOV     #1,TSTFLAG              ;TEST PATTERN FLAG
041742
041742  012737  000002  002310          100$:   MOV     #2,COUNT                ;GET FIRST BYTE COUNT
041750
                                        ;       Do a Write Subsystem WRITE NPR to set tape direction out
041750  012700  000100                  ;       MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
041754  004737  047376                          JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
041760  012704  047720                          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
041764  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
041770  004737  016336                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
041774                                          FORCERROR       102$            ;@@FORCE ERROR IF FORCER=1
042010  103407                          BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
042012  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
042014                                  NEXT.ERRNO
042014                          102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
042014  104455                                                                  TRAP    C$ERDF
042016  001152                                                                  .WORD   618
042020  046343                                                                  .WORD   T174SSR
042022  012046                                                                  .WORD   PKTSSR
042024  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
042030                          105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
042030  104406                                                                  TRAP    C$CLP1
                                        ;       Do a Write Subsystem WRITE FIFO
042032  004737  047476                  ;       JSR     PC,T17CLEXP             ;CLEAR EXPD BUFFER
042036  012701  046054                          MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
042042  013702  002310                          MOV     COUNT,R2                ;TEST PATTERN SIZE
042046  022737  000001  002314                  CMP     #1,TSTFLAG              ;INCREMENT PATTERN THIS TIME THRU?
042054  001005                          BNE     115$                    ;BR IF NO
042056  005000                          CLR     R0                      ;INCREMENT TEST PATTERN
042060  110021                  110$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
042062  005200                          INC     R0                      ;SET NEXT PATTERN
042064  005302                          DEC     R2                      ;DONE?
042066  003374                          BGT     110$                    ;BR IF NO
042070  022737  000002  002314  115$:   CMP     #2,TSTFLAG              ;DECREMENT PATTERN THIS TIME THRU?
042076  001006                          BNE     125$                    ;BR IF NO
042100  012700  000377                  MOV     #377,R0                 ;DECREMENT TEST PATTERN
042104  110021                  120$:   MOVB    R0,(R1)+                ;STORE DECREMENT TEST BYTE
042106  005300                          DEC     R0                      ;SET NEXT PATTERN
042110  005302                          DEC     R2                      ;DONE?
042112  003374                          BGT     120$                    ;BR IF NO
042114  022737  000003  002314  125$:   CMP     #3,TSTFLAG              ;TSTBLK PATTERNS THIS TIME THRU?
042122  001005                          BNE     135$                    ;BR IF NO
042124  012700  002752                  MOV     #TSTBLK,R0              ;FLOAT 1'S/0'S ETC. TEST TABLE
042130  112021                  130$:   MOVB    (R0)+,(R1)+             ;STORE A TSTBLK BYTE
042132  005302                          DEC     R2                      ;DONE?
042134  003375                          BGT     130$                    ;BR IF NO
042136                          135$:
042136  013700  002310                  MOV     COUNT,R0                ;FIFO BYTE COUNT
042142  012701  046054                  MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
042146  004737  047422                  JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
042152  012704  047720                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
042156  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
6674 042162  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6675 042166                                  FORCERROR       142$                    ;@@FORCE ERROR IF FORCER=1
6676 042202  103407                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
6677 042204  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6678 042206                                  NEXT.ERRNO
6679 042206                          142$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042206  104455                                                                  TRAP    C$ERDF
     042210  001153                                                                  .WORD   619
     042212  046406                                                                  .WORD   T175SSR
     042214  012046                                                                  .WORD   PKTSSR
6680 042216  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
6681 042222                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042222  104406                                                                  TRAP    C$CLP1
6682
6683                                 ;       Do a Write Subsystem READ STATUS
6684 042224  004737  047334                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6685 042230  012704  047720                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6686 042234  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6687 042240  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6688 042244                                  FORCERROR       157$                    ;@@FORCE ERROR IF FORCER=1
6689 042260  103407                          BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
6690 042262  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6691 042264                                  NEXT.ERRNO
6692 042264                          157$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042264  104455                                                                  TRAP    C$ERDF
     042266  001154                                                                  .WORD   620
     042270  046276                                                                  .WORD   T173SSR
     042272  012046                                                                  .WORD   PKTSSR
6693 042274  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
6694 042300                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042300  104406                                                                  TRAP    C$CLP1
6695
6696                                 ;       Set WORDS 0 7 of e pd message buffer = to recv since not testing
6697 042302  004737  047516                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
6698 042306  012701  045752                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
6699 042312  012702  047612                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
6700 042316  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
6701 042320  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
6702 042322  052711  000020                  BIS     #S2.INPDY,(R1)          ;SET EXP INPUT READY= 1
6703 042326  052711  000040                  BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
6704 042332  042711  000200                  BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
6705 042336  042711  000100                  BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
6706                                 ;       If Input Ready NOT=1 then Print Error
6707                                 ;       If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
6708 042342  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6709 042344  012701  047572                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
6710 042350  012702  045732                  MOV     #T17EXP,R2              ;EXPD ADDRESS
6711 042354  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
6712 042360  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6713 042364                                  FORCERROR       162$,NOTSSR             ;@@D
6714 042374  103404                          BCS     170$                    ;BR IF YES
6715 042376
6716 042376                          162$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     042376  104456                                                                  TRAP    C$ERHRD
     042400  001155                                                                  .WORD   621
     042402  046673                                                                  .WORD   T173CMP
     042404  012350                                                                  .WORD   MSGSTAT
```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
6717 042406                          170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042406  104406                                                                  TRAP    C$CLP1
6718
6719                                 ;       Do Write Subsystem READ FIFO
6720 042410  013700  002310                  MOV     COUNT,R0                ;SET READ BYTE COUNT
6721 042414  004737  047456                  JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
6722 042420  012704  047720                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6723 042424  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6724 042430  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6725 042434                                  FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
6726 042450  103407                          BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
6727 042452  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6728 042454                                  NEXT.ERRNO
6729 042454                          172$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042454  104455                                                                  TRAP    C$ERDF
     042456  001156                                                                  .WORD   622
     042460  046452                                                                  .WORD   T176SSR
     042462  012046                                                                  .WORD   PKTSSR
6730 042464  C05237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
6731 042470                          180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042470  104406                                                                  TRAP    C$CLP1
6732
6733                                 ;       If Data read from FIFO NOT= to Data sent Then Print Error
6734 042472  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6735 042474  012702  046054                  MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
6736 042500  012701  047612                  MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
6737 042504  013703  002310                  MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
6738 042510  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6739 042514                                  FORCERROR       192$,NOTSSR             ;@@D
6740 042524  103406                          BCS     200$                    ;BR IF YES
6741 042526                                  NEXT.ERRNO
6742 042526  013701  002310          192$:   MOV     COUNT,R1                ;GET BYTE COUNT
6743 042532                                  ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
     042532  104456                                                                  TRAP    C$ERHRD
     042534  001157                                                                  .WORD   623
     042536  047042                                                                  .WORD   T175CMP
     042540  012170                                                                  .WORD   FIFEXP
6744 042542                          200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042542  104406                                                                  TRAP    C$CLP1
6745
6746                                 ;       Do a Write Subsystem READ STATUS
6747 042544  004737  047334                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6748 042550  012704  047720                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6749 042554  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6750 042560  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6751 042564                                  FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
6752 042600  103407                          BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
6753 042602  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6754 042604                                  NEXT.ERRNO
6755 042604                          212$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042604  104455                                                                  TRAP    C$ERDF
     042606  001160                                                                  .WORD   624
     042610  046276                                                                  .WORD   T173SSR
     042612  012046                                                                  .WORD   PKTSSR
6756 042614  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
6757 042620                          220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042620  104406                                                                  TRAP    C$CLP1
```

```
6758                                    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
6759 042622  004737  047516                     JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
6760 042626  012701  045752                     MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
6761 042632  012702  047612                     MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
6762 042636  012221                             MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
6763 042640  011211                             MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
6764 042642  052711  000020                     BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
6765 042646  042711  000040                     BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
6766 042652  042711  000200                     BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
6767 042656  042711  000100                     BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
6768                                    ;       If Input Ready NOT=1 then Print Error
6769                                    ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6770 042662  005000                             CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6771 042664  012701  047572                     MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
6772 042670  012702  045732                     MOV     #T17EXP,R2              ;EXPD ADDRESS
6773 042674  012703  000024                     MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
6774 042700  004737  011500                     JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6775 042704                                     FORCERROR       232$,NOTSSR     ;@@D
6776 042714  103404                             BCS     240$                    ;BR IF YES
6777 042716                                     NEXT.ERRNO
6778 042716                             232$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     042716  104456                                                                    TRAP    C$ERHRD
     042720  001161                                                                    .WORD   625
     042722  046757                                                                    .WORD   T174CMP
     042724  012350                                                                    .WORD   MSGSTAT
6779 042726                             240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042726  104406                                                                    TRAP    C$CLP1
6780 042730                                     FORCEXIT        250$            ;@@D
6781 042740  005237  002310                     INC     COUNT                   ;GET NEXT BYTE COUNT
6782 042744  023727  002310  000077             CMP     COUNT,#77               ;DONE 0 TO 77
6783 042752  101002                             BHI     250$                    ;BR IF YES
6784 042754  000137  041750                     JMP     100$                    ;DO ANOTHER BYTE COUNT
6785 042760  005237  002314             250$:   INC     TSTFLAG                 ;GET NEXT TEST PATTERN CODE
6786 042764  023727  002314  000003             CMP     TSTFLAG,#3              ;DONE INC,DEC,TSTBLK PATTERNS?
6787 042772  101002                             BHI     255$                    ;BR IF YES
6788 042774  000137  041742                     JMP     95$                     ;DO ANOTHER TEST PATTERN
6789 043000                             255$:
6790 043000                                     ENDSUB                          ;////////////// END SUBTEST //////////
     043000                                                                            L10061:
     043000  104403                                                                            TRAP    C$ESUB
6791
6792 043002  005737  002222                     TST     FATFLG                  ;ANY FATAL ERRORS ?
6793 043006  001402                             BEQ     260$                    ;BRANCH IF NOT
6794 043010  004737  017202                     JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
6795 043014                             260$:
6796
6797
6798
6799                                            .SBTTL  TEST  6: SUBTEST 4: FIFO Verify ILW Status
6800
6801                                    ;**
6802                                    ; TEST 6: SUBTEST 4:
6803                                    ;
6804                                    ; SUBTEST DESCRIPTION:
6805                                    ;
6806                                    ;       This subtest verifies that reading the FIFO when it is
6807                                    ;       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
                                ;--
043014                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
043014                                                                                  T6.4:
043014  104402                                                                                 TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
043016                          5$:
043016  004737  015774                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
043022  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
043024  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043026                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
043026  104455                                                                          TRAP    C$ERDF
043030  001161                                                                          .WORD   625
043032  003652                                                                          .WORD   SFIERR
043034  012034                                                                          .WORD   SFIMSG

                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
043036  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
043042  012704  047550                  MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
043046  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
043052                                  FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
043066  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
043070  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043072                                  NEXT.ERRNO
043072                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
043072  104455                                                                          TRAP    C$ERDF
043074  001162                                                                          .WORD   626
043076  046175                                                                          .WORD   T17SSR
043100  012046                                                                          .WORD   PKTSSR
043102  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
043106                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
043106  104406                                                                          TRAP    C$CLP1

                                ;       Do Write Subsystem READ FIFO with byte count equal to 1
043110  012700  000001                  MOV     #1,R0                   ;SET READ BYTE COUNT
043114  004737  047456                  JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
043120  012704  047720                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
043124  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
043130  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043134                                  FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
043150  103407                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
043152  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043154                                  NEXT.ERRNO
043154                          142$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
043154  104455                                                                          TRAP    C$ERDF
043156  001163                                                                          .WORD   627
```

TEST 6: SUBTEST 4: FIFO Verify ILW Status

```
      043160  046452                                                          .WORD   T176SSR
      043162  012046                                                          .WORD   PKTSSR
 6852 043164  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 6853 043170                     150$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      043170  104406                                                          TRAP    C$CLP1
 6854
 6855                            ;      Do a Write Subsystem READ STATUS
 6856 043172  004737  047334            JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
 6857 043176  012704  047720            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 6858 043202  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 6859 043206  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6860 043212                            FORCERROR       162$                    ;@@FORCE ERROR IF FORCER=1
 6861 043226  103407                    BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
 6862 043230  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6863 043232                            NEXT.ERRNO
 6864 043232                     162$:  ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      043232  104455                                                          TRAP    C$ERDF
      043234  001164                                                          .WORD   628
      043236  046276                                                          .WORD   T173SSR
      043240  012046                                                          .WORD   PKTSSR
 6865 043242  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 6866 043246                     170$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      043246  104406                                                          TRAP    C$CLP1
 6867                            ;      Set WORDS 0-7 of expd message buffer = to recv since not testing
 6868 043250  004737  047516            JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
 6869 043254  012701  045752            MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 6870 043260  012702  047612            MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 6871 043264  012221                    MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 6872 043266  011211                    MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 6873 043270  052711  000020            BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
 6874 043274  042711  000040            BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
 6875 043300  042711  000200            BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
 6876 043304  052711  000100            BIS     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=1
 6877                            ;      If Input Ready NOT=1 then Print Error
 6878                            ;      If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
 6879                            ;      If Last Word (ILW) NOT=1 Then Print Error
 6880 043310  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 6881 043312  012701  047572            MOV     #T178FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 6882 043316  012702  045732            MOV     #T17EXP,R2              ;EXPD ADDRESS
 6883 043322  012703  000024            MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 6884 043326  004737  011500            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 6885 043332                            FORCERROR       172$,NOTSSR     ;@@D
 6886 043342  103404                    BCS     180$                    ;BR IF YES
 6887 043344                            NEXT.ERRNO
 6888 043344                     172$:  ERRHRD  ERRNO,T176CMP,MSGSTAT   ;REPORT ERROR
      043344  104456                                                          TRAP    C$ERHRD
      043346  001165                                                          .WORD   629
      043350  047116                                                          .WORD   T176CMP
      043352  012350                                                          .WORD   MSGSTAT
 6889 043354                     180$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      043354  104406                                                          TRAP    C$CLP1
 6890
 6891 043356                            ENDSUB                          ;////////////// END SUBTEST //////////
      043356                                                                  L10062:
      043356  104403                                                          TRAP    C$ESUB
 6892
 6893 043360  005737  002222            TST     FATFLG                  ;ANY FATAL ERRORS ?
```

TEST 6: SUBTEST 4: FIFO Verify ILW Status

```
6894 043364  001402                    BEQ     260$                    ;BRANCH IF NOT
6895 043366  004737  017202            JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
6896 043372                        260$:
6897
6898
6899                                    .SBTTL  TEST  6: SUBTEST 5: FIFO Verify Input Ready
6900
6901                               ;**
6902                               ; TEST 6: SUBTEST 5:
6903                               ;
6904                               ; SUBTEST DESCRIPTION:
6905                               ;
6906                               ;       This subtest verifies that writing 64. bytes into the FIFO
6907                               ;       without reading any out causes the Input Ready status to
6908                               ;       negate. The Subtest then verfiies that writing a 65th byte
6909                               ;       into the FIFO causes the Data In Miss status to assert.
6910                               ;       Next it is verified that the original 64 bytes can be read
6911                               ;       out correctly and that the data has not been corrupted.
6912                               ;
6913                               ; TEST STEPS:
6914                               ;
6915                               ; BEGIN
6916                               ;       Write to TSSR to soft initialize
6917                               ;       Do a WRITE CHARACTERISTICS to setup a message buffer
6918                               ;       Do a Write Subsystem WRITE NPR to set tape direction out
6919                               ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
6920                               ;       Do a Write Subsystem READ STATUS
6921                               ;       If Input Ready  NOT=0   Then Print Error
6922                               ;       If Output Ready NOT=1   Then Print Error
6923                               ;       If Data In Miss NOT=0   Then Print Error
6924                               ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
6925                               ;       Do a Write Subsystem READ STATUS
6926                               ;       If Input Ready  NOT=0   Then Print Error
6927                               ;       If Output Ready NOT=1   Then Print Error
6928                               ;       If Data In Miss NOT=1   Then Print Error
6929                               ;       Do Write Subsystem READ FIFO
6930                               ;       If Data read from FIFO NOT= to Data sent Then Print Error
6931                               ;       Do a Write Subsystem READ STATUS
6932                               ;       If Input Ready  NOT=1   Then Print Error
6933                               ;       If Output Ready NOT=0   Then Print Error
6934                               ;       If Data In Miss NOT=1   Then Print Error
6935                               ;  END
6936                               ;--
6937 043372                            BGNSUB                          ;////////////// BEGIN SUBTEST //////////
     043372                                                                          T6.5:
     043372  104402                                                                    TRAP    L$BSUB
6938
6939                               ;       Write to TSSR register to soft initialize the controller
6940 043374                        5$:
6941 043374  004737  015774            JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
6942 043400  103405                    BCS     10$                     ;BR IF SOFT INIT OKAY
6943 043402  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6944 043404                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     043404  104455                                                                    TRAP    C$ERDF
     043406  001165                                                                    .WORD   629
     043410  003652                                                                    .WORD   SFIERR
     043412  012034                                                                    .WORD   SFIMSG
```

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```
6945                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
6946 043414  005037  002222             10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
6947 043420  012704  047550                     MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
6948 043424  004737  010662                     JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
6949 043430                                     FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
6950 043444  103407                           BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
6951 043446  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6952 043450                                     NEXT.ERRNO
6953 043450                             42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     043450  104455                                                                  TRAP    C$ERDF
     043452  001166                                                                  .WORD   630
     043454  046175                                                                  .WORD   T17SSR
     043456  012046                                                                  .WORD   PKTSSR
6954 043460  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
6955 043464                             50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043464  104406                                                                  TRAP    C$CLP1
6956
6957                                    ;       Do a Write Subsystem WRITE NPR to set tape direction out
6958 043466  C12700  000100             100$:   MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
6959 043472  004737  047376                     JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
6960 043476  012704  047720                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6961 043502  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6962 043506  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6963 043512                                     FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
6964 043526  103407                           BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
6965 043530  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6966 043532                                     NEXT.ERRNO
6967 043532                             102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043532  104455                                                                  TRAP    C$ERDF
     043534  001167                                                                  .WORD   631
     043536  046343                                                                  .WORD   T174SSR
     043540  012046                                                                  .WORD   PKTSSR
6968 043542  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
6969 043546                             105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043546  104406                                                                  TRAP    C$CLP1
6970
6971                                    ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
6972 043550  012737  000100  002310             MOV     #64.,COUNT              ;WRITE 64 BYTES
6973 043556  012701  046054                     MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
6974 043562  012702  000100                     MOV     #64.,R2                 ;TEST PATTERN SIZE
6975 043566  005000                           CLR     R0                      ;INCREMENT TEST PATTERN
6976 043570  110021                     110$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
6977 043572  005200                           INC     R0                      ;SET NEXT PATTERN
6978 043574  005302                           DEC     R2                      ;DONE?
6979 043576  003374                           BGT     110$                    ;BR IF NO
6980 043600  013700  002310                   MOV     COUNT,R0                ;FIFO BYTE COUNT
6981 043604  012701  046054                   MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
6982 043610  004737  047422                   JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
6983 043614  012704  047720                   MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6984 043620  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6985 043624  004737  016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6986 043630                                   FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
6987 043644  103407                         BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
6988 043646  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6989 043650                                   NEXT.ERRNO
6990 043650                           142$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043650  104455                                                                  TRAP    C$ERDF
```

```
        043652  001170                                                          .WORD   632
        043654  046406                                                          .WORD   T175SSR
        043656  012046                                                          .WORD   PKTSSR
   6991 043660  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
   6992 043664                        150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        043664  104406                                                          TRAP    C$CLP1
   6993
   6994                                ;       Do a Write Subsystem READ STATUS
   6995                                ;       If Input Ready  NOT=0   Then Print Error
   6996                                ;       If Output Ready NOT=1   Then Print Error
   6997                                ;       If Data In Miss NOT=0   Then Print Error
   6998 043666  004737  047334                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
   6999 043672  012704  047720                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
   7000 043676  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
   7001 043702  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
   7002 043706                                FORCERROR       157$            ;@@DFORCE ERROR IF FORCER=1
   7003 043722  103407                        BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
   7004 043724  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
   7005 043726                                NEXT.ERRNO
   7006 043726                        157$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        043726  104455                                                          TRAP    C$ERDF
        043730  001171                                                          .WORD   633
        043732  046276                                                          .WORD   T173SSR
        043734  012046                                                          .WORD   PKTSSR
   7007 043736  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
   7008 043742                        160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        043742  104406                                                          TRAP    C$CLP1
   7009                                ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
   7010 043744  004737  047516                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
   7011 043750  012701  045752                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
   7012 043754  012702  047612                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
   7013 043760  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
   7014 043762  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
   7015 043764  042711  000020                BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
   7016 043770  052711  000040                BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
   7017 043774  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
   7018 044000  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
   7019 044002  012701  047572                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
   7020 044006  012702  045732                MOV     #T17EXP,R2              ;EXPD ADDRESS
   7021 044012  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
   7022 044016  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
   7023 044022                                FORCERROR       162$,NOTSSR     ;@@D
   7024 044032  103404                        BCS     170$                    ;BR IF YES
   7025 044034                                NEXT.ERRNO
   7026 044034                        162$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
        044034  104456                                                          TRAP    C$ERHRD
        044036  001172                                                          .WORD   634
        044040  046673                                                          .WORD   T173CMP
        044042  012350                                                          .WORD   MSGSTAT
   7027 044044                        170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        044044  104406                                                          TRAP    C$CLP1
   7028
   7029
   7030                                ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
   7031 044046  012700  000001                MOV     #1,R0                   ;FIFO BYTE COUNT
   7032 044052  012701  046054                MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
   7033 044056  004737  047422                JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
```

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```
7034 044062  012704  047720                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7035 044066  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7036 044072  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7037 044076                                FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
7038 044112  103407                        BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
7039 044114  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7040 044116                                NEXT.ERRNO
7041 044116                        172$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044116  104455                                                                  TRAP    C$ERDF
     044120  001173                                                                  .WORD   635
     044122  046406                                                                  .WORD   T175SSR
     044124  012046                                                                  .WORD   PKTSSR
7042 044126  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
7043 044132                        180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044132  104406                                                                  TRAP    C$CLP1
7044
7045                               ;       Do a Write Subsystem READ STATUS
7046                               ;       If Input Ready  NOT=0   Then Print Error
7047                               ;       If Output Ready NOT=1   Then Print Error
7048                               ;       If Data In Miss NOT=1   Then Print Error
7049 044134  004737  047334                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
7050 044140  012704  047720                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7051 044144  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7052 044150  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7053 044154                                FORCERROR       187$                    ;@@DFORCE ERROR IF FORCER=1
7054 044170  103407                        BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
7055 044172  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7056 044174                                NEXT.ERRNO
7057 044174                        187$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044174  104455                                                                  TRAP    C$ERDF
     044176  001174                                                                  .WORD   636
     044200  046276                                                                  .WORD   T173SSR
     044202  012046                                                                  .WORD   PKTSSR
7058 044204  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
7059 044210                        190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044210  104406                                                                  TRAP    C$CLP1
7060                               ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
7061 044212  004737  047516                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
7062 044216  012701  045752                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
7063 044222  012702  047612                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
7064 044226  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
7065 044230  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
7066 044232  042711  000020                BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
7067 044236  052711  000040                BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
7068 044242  052711  000200                BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
7069 044246  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
7070 044250  012701  047572                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
7071 044254  012702  045732                MOV     #T17EXP,R2              ;EXPD ADDRESS
7072 044260  012703  0000..                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
7073 044264  004737  01150)                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
7074 044270                                FORCERROR       192$,NOTSSR             ;@@D
7075 044300  103404                        BCS     200$                    ;BR IF YES
7076 044302                                NEXT.ERRNO
7077 044302                        192$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     044302  104456                                                                  TRAP    C$ERHRD
     044304  001175                                                                  .WORD   637
     044306  046673                                                                  .WORD   T173CMP
```

```
                                                                                    .WORD   MSGSTAT
7078 044312                         200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044312 104406                                                                  TRAP    C$CLP1
7079                                ;       Do Write Subsystem READ FIFO
7080 044314 013700 002310                   MOV     COUNT,R0                ;SET READ BYTE COUNT
7081 044320 004737 047456                   JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
7082 044324 012704 047720                   MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7083 044330 010465 000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7084 044334 004737 016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7085 044340                                 FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
7086 044354 103407                          BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
7087 044356 010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7088 044360                                 NEXT.ERRNO
7089 044360                         212$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044360 104455                                                                  TRAP    C$ERDF
     044362 001176                                                                  .WORD   638
     044364 046452                                                                  .WORD   T176SSR
     044366 012046                                                                  .WORD   PKTSSR
7090 044370 C05237 002222                   INC     FATFLG                  ;SET FATAL ERROR FLAG
7091 044374                         220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044374 104406                                                                  TRAP    C$CLP1
7092
7093                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
7094 044376 005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
7095 044400 012702 046054                   MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
7096 044404 012701 047612                   MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
7097 044410 013703 002310                   MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
7098 044414 004737 011500                   JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
7099 044420                                 FORCERROR       232$,NOTSSR     ;@@D
7100 044430 103406                          BCS     240$                    ;BR IF YES
7101 044432                                 NEXT.ERRNO
7102 044432 013701 002310           232$:   MOV     COUNT,R1                ;GET BYTE COUNT
7103 044436                                 ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
     044436 104456                                                                  TRAP    C$ERHRD
     044440 001177                                                                  .WORD   639
     044442 047042                                                                  .WORD   T175CMP
     044444 012170                                                                  .WORD   FIFEXP
7104 044446                         240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044446 104406                                                                  TRAP    C$CLP1
7105
7106                                ;       Do a Write Subsystem READ STATUS
7107                                ;       If Input Ready  NOT=1   Then Print Error
7108                                ;       If Output Ready NOT=0   Then Print Error
7109                                ;       If Data In Miss NOT=1   Then Print Error
7110 044450 004737 047334                   JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
7111 044454 012704 047720                   MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7112 044460 010465 000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7113 044464 004737 016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7114 044470                                 FORCERROR       252$                    ;@@DFORCE ERROR IF FORCER=1
7115 044504 103407                          BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
7116 044506 010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7117 044510                                 NEXT.ERRNO
7118 044510                         252$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044510 104455                                                                  TRAP    C$ERDF
     044512 001200                                                                  .WORD   640
     044514 046276                                                                  .WORD   T173SSR
     044516 012046                                                                  .WORD   PKTSSR
```

```
7119 044520  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
7120 044524                             260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044524  104406                                                                    TRAP    C$CLP1
7121                                    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
7122 044526  004737  047516                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
7123 044532  012701  045752                    MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
7124 044536  012702  047612                    MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
7125 044542  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
7126 044544  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
7127 044546  052711  000020                    BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
7128 044552  042711  000040                    BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
7129 044556  052711  000200                    BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
7130 044562  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
7131 044564  012701  047572                    MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
7132 044570  012702  045732                    MOV     #T17EXP,R2              ;EXPD ADDRESS
7133 044574  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
7134 044600  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
7135 044604                                    FORCERROR       272$,NOTSSR     ;###
7136 044614  103404                            BCS     280$                    ;BR IF YES
7137 044616                                    NEXT.ERRNO
7138 044616                             272$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     044616  104456                                                                    TRAP    C$ERHRD
     044620  001201                                                                    .WORD   641
     044622  046757                                                                    .WORD   T174CMP
     044624  012350                                                                    .WORD   MSGSTAT
7139 044626                             280$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044626  104406                                                                    TRAP    C$CLP1
7140
7141 044630                                    ENDSUB                          ;////////////// END SUBTEST //////////
     044630                                                                         L10063:
     044630  104403                                                                    TRAP    C$ESUB
7142
7143 044632  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
7144 044636  001402                            BEQ     300$                    ;BRANCH IF NOT
7145 044640  004737  017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
7146 044644                             300$:
7147
7148
7149
7150                                            .SBTTL  TEST  6: SUBTEST 6: FIFO Verify Reset FIFO Test
7151
7152                                    ;++
7153                                    ; TEST 6: SUBTEST 6:
7154                                    ;
7155                                    ; SUBTEST DESCRIPTION:
7156                                    ;
7157                                    ;       This subtest verifies that the Reset FIFO function within
7158                                    ;       the Write Miscellaneous Control 1 function initializes
7159                                    ;       the FIFO to correct intial status. The following steps
7160                                    ;       are performed:
7161                                    ;               1. Reset an already initialized FIFO and check for
7162                                    ;                  proper status.
7163                                    ;               2. Write a varying number of bytes (1-65.) into the
7164                                    ;                  FIFO and verify that after each block of bytes is
7165                                    ;                  written the FIFO can be be reset to it's initial
7166                                    ;                  state.
7167                                    ;
```

```
                                ; 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 M sc 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
                                ;--
044644                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
044644                                                                                  T6.6:
044644  104402                                                                                  TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
                                5$:
044646  004737  015774                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
044652  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
044654  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
044656                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
044656  104455                                                                          TRAP    C$ERDF
044660  001201                                                                          .WORD   641
044662  003652                                                                          .WORD   SFIERR
044664  012034                                                                          .WORD   SFIMSG

                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
044666  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
044672  012704  047550                  MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
044676  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
044702                                  FORCERROR       42$                     ;BBDFORCE ERROR IF FORCER=1
044716  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
044720  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
044722                                  NEXT.ERRNO
044722                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
044722  104455                                                                          TRAP    C$ERDF
044724  001202                                                                          .WORD   642
044726  046175                                                                          .WORD   T17SSR
044730  012046                                                                          .WORD   PKTSSR
044732  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
044736                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
044736  104406                                                                          TRAP    C$CLP1

                                ;       Do a Write Subsystem Write Misc to Reset FIFO
044740  004737  047354                  JSR     PC,T17RSFIF             ;SETUP PKT FOR WRITE MISC RESET FIFO
044744  012704  047720                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
044750  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
044754  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
044760                                  FORCERROR       62$                     ;BBDFORCE ERROR IF FORCER=1
044774  103407                          BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```
 7214 044776  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7215 045000                                NEXT.ERRNO
 7216 045000                        6$:     ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      045000  104455                                                          TRAP    C$ERDF
      045002  001203                                                          .WORD   643
      045004  046232                                                          .WORD   T172SSR
      045006  012046                                                          .WORD   PKTSSR
 7217 045010  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 7218 045014                        70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      045014  104406                                                          TRAP    C$CLP1
 7219
 7220                               ;       Do a Write Subsystem READ STATUS
 7221                               ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
 7222                               ;         signals NOT=0 Then Print Error
 7223 045016  004737  047334                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
 7224 045022  012704  047720                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7225 045026  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7226 045032  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7227 045036                                FORCERROR       77$             ;@@FORCE ERROR IF FORCER=1
 7228 045052  103407                        BCS     80$                     ;BR IF CARRY SET (GOOD RETURN)
 7229 045054  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7230 045056                                NEXT.ERRNO
 7231 045056                        77$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      045056  104455                                                          TRAP    C$ERDF
      045060  001204                                                          .WORD   644
      045062  046276                                                          .WORD   T173SSR
      045064  012046                                                          .WORD   PKTSSR
 7232 045066  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 7233 045072                        80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      045072  104406                                                          TRAP    C$CLP1
 7234 045074  004737  047516                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 7235 045100  012701  045752                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 7236 045104  012702  047612                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 7237 045110  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
 7238 045112  042711  002000                BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
 7239 045116  042711  001000                BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
 7240 045122  042711  000400                BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
 7241 045126  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
 7242 045134  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 7243 045136  012701  047572                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 7244 045142  012702  045732                MOV     #T17EXP,R2              ;EXPD ADDRESS
 7245 045146  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 7246 045152  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 7247 045156                                FORCERROR       92$,NOTSSR      ;@@D
 7248 045166  103404                        BCS     100$                    ;BR IF YES
 7249 045170                                NEXT.ERRNO
 7250 045170                        92$:    ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
      045170  104456                                                          TRAP    C$ERHRD
      045172  001205                                                          .WORD   645
      045174  047224                                                          .WORD   T177CMP
      045176  012350                                                          .WORD   MSGSTAT
 7251 045200                        100$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      045200  104406                                                          TRAP    C$CLP1
 7252
 7253                               ;       Do a Write Subsystem WRITE NPR to set tape direction out
 7254 045202  012700  000100                MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
 7255 045206  004737  047376                JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```
7256 045212  012704  047720                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7257 045216  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7258 045222  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7259 045226                                FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
7260 045242  103407                        BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
7261 045244  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7262 045246                                NEXT.ERRNO
7263 045246                        112$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045246  104455                                                                  TRAP    C$ERDF
     045250  001206                                                                  .WORD   646
     045252  046343                                                                  .WORD   T174SSR
     045254  012046                                                                  .WORD   PKTSSR
7264 045256  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
7265 045262                        120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045262  104406                                                                  TRAP    C$CLP1
7266
7267                               ;       Setup incrementing pattern in FIFO data buffer
7268 045264  012701  045752                MOV     #T17EXSTA,R1            ;EXPD WRITE FIFO DATA BUFFER
7269 045270  C12702  000100                MOV     #64.,R2                 ;TEST PATTERN SIZE
7270 045274  005000                        CLR     R0                      ;INCREMENT TEST PATTERN
7271 045276  110021                130$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
7272 045300  005200                        INC     R0                      ;SET NEXT PATTERN
7273 045302  005302                        DEC     R2                      ;DONE?
7274 045304  003374                        BGT     130$                    ;BR IF NO
7275
7276                               ;  REPEAT FOR BYTE COUNT 1 TO 65.
7277 045306  012737  000001  002310        MOV     #1,COUNT                ;GET FIRST BYTE COUNT
7278                               ;       Do a Write Subsystem WRITE FIFO with the current byte count
7279 045314                        150$:                                   ;REPEAT LOOP LABEL
7280 045314  013700  002310                MOV     COUNT,R0                ;FIFO BYTE COUNT
7281 045320  012701  045752                MOV     #T17EXSTA,R1            ;FIFO WRITE DATA ADDRESS
7282 045324  004737  047422                JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
7283 045330  012704  047720                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7284 045334  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7285 045340  004737  016336                JSR     PC,CHKTSSP              ;WAIT FOR SSR TO SET
7286 045344                                FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
7287 045360  103407                        BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
7288 045362  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7289 045364                                NEXT.ERRNO
7290 045364                        152$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045364  104455                                                                  TRAP    C$ERDF
     045366  001207                                                                  .WORD   647
     045370  046406                                                                  .WORD   T175SSR
     045372  012046                                                                  .WORD   PKTSSR
7291 045374  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
7292 045400                        160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045400  104406                                                                  TRAP    C$CLP1
7293
7294                               ;       Do a Write Subsystem Write Misc to Reset FIFO
7295 045402  004737  047354                JSR     PC,T17RSFIF             ;SETUP PKT FOR WRITE MISC RESET FIFO
7296 045406  012704  047720                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7297 045412  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7298 045416  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7299 045422                                FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
7300 045436  103407                        BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
7301 045440  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7302 045442                                NEXT.ERRNO
```

```
7303 045442                          162$:   ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045442  104455                                                                TRAP    C$ERDF
     045444  001210                                                                .WORD   648
     045446  046232                                                                .WORD   T172SSR
     045450  012046                                                                .WORD   PKTSSR
7304 045452  005237  002222                  INC     FATFLG                      ;SET FATAL ERROR FLAG
7305 045456                          170$:   CKLOOP                              ;LOOP ON ERROR, IF FLAG SET
     045456  104406                                                                TRAP    C$CLP1
7306
7307                                 ;       Do a Write Subsystem READ STATUS
7308                                 ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7309                                 ;         signals NOT=0 Then Print Error
7310 045460  004737  047334                  JSR     PC,T17SRD                   ;SETUP PACKET FOR READ STATUS
7311 045464  012704  047720                  MOV     #T17PK2,R4                  ;GET WRITE SUBSYSTEM COMMAND PACKET
7312 045470  010465  000000                  MOV     R4,TSDB(R5)                 ;SET THE PACKET ADDRESS TO EXECUTE
7313 045474  004737  016336                  JSR     PC,CHKTSSR                  ;WAIT FOR SSR TO SET
7314 045500                                  FORCERROR       177$                        ;@@FORCE ERROR IF FORCER=1
7315 045514  103407                          BCS     180$                        ;BR IF CARRY SET (GOOD RETURN)
7316 045516  C10001                          MOV     R0,R1                       ;SAVE CONTENTS OF TSSR
7317 045520                                  NEXT.ERRNO
7318 045520                          177$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045520  104455                                                                TRAP    C$ERDF
     045522  001211                                                                .WORD   649
     045524  046276                                                                .WORD   T173SSR
     045526  012046                                                                .WORD   PKTSSR
7319 045530  005237  002222                  INC     FATFLG                      ;SET FATAL ERROR FLAG
7320 045534                          180$:   CKLOOP                              ;LOOP ON ERROR, IF FLAG SET
     045534  104406                                                                TRAP    C$CLP1
7321 045536  004737  047516                  JSR     PC,T17SETEXP                ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
7322 045542  012701  045752                  MOV     #T17EXSTA,R1                ;GET EXPECTED READ STATUS
7323 045546  012702  047612                  MOV     #T17BFSTA,R2                ;GET RECV READ STATUS
7324 045552  011211                          MOV     (R2),(R1)                   ;SET EXPD WORD #8 = RECV TEMP
7325 045554  042711  002000                  BIC     #S1.ICER,(R1)               ;SET EXPD ICER  =0
7326 045560  042711  001000                  BIC     #S1.IFMK,(R1)               ;SET EXPD IFMK  =0
7327 045564  042711  000400                  BIC     #S1.IHER,(R1)               ;SET EXPD IHER  =0
7328 045570  016261  000002  000002          MOV     2(R2),2(R1)                 ;SET EXPD WORD #9 = RECV (NOT TESTING)
7329 045576  005000                          CLR     R0                          ;HIGH RECV ADDRESS FOR CKMSG2
7330 045600  012701  047572                  MOV     #T17BFR,R1                  ;LOW RECV ADDRESS FOR CKMSG2
7331 045604  012702  045732                  MOV     #T17EXP,R2                  ;EXPD ADDRESS
7332 045610  012703  000024                  MOV     #20.,R3                     ;NUMBER OF BYTES TO COMPARE
7333 045614  004737  011500                  JSR     PC,CKMSG2                   ;EXPD EQUAL RECV?
7334 045620                                  FORCERROR       192$,NOTSSR         ;@@D
7335 045630  103404                          BCS     200$                        ;BR IF YES
7336 045632                                  NEXT.ERRNO
7337 045632                          192$:   ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
     045632  104456                                                                TRAP    C$ERHRD
     045634  001212                                                                .WORD   650
     045636  047224                                                                .WORD   T177CMP
     045640  012350                                                                .WORD   MSGSTAT
7338 045642                          200$:   CKLOOP                              ;LOOP ON ERROR, IF FLAG SET
     045642  104406                                                                TRAP    C$CLP1
7339
7340
7341 045644                          250$:
7342 045644                                  FORCEXIT        260$
7343 045654  005237  002310                  INC     COUNT                       ;GET NEXT BYTE COUNT
7344 045660  023727  002310  000101          CMP     COUNT,#65.                  ;DONE ALL BYTES?
```

```
7345 045666  101002                          BHI     260$                    ;BR IF YES
7346 045670  000137  045314                  JMP     150$                    ;DO ANOTHER BYTE COUNT
7347 045674                          260$:
7348
7349 045674                                  ENDSUB                          ;////////////// END SUBTEST //////////
     045674                                                                                  L10064:
     045674  104403                                                                                  TRAP    C$ESUB
7350
7351 045676  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
7352 045702  001402                          BEQ     300$                    ;BRANCH IF NOT
7353 045704  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
7354 045710  004737  016456          300$:   JSR     PC,TSTLOOP              ;DO ITERATIONS?
7355 045714  103002                          BCC     305$                    ;BR IF NO
7356 045716  000137  037754                  JMP     T17LOOP                 ;LOOP UNTIL ITERATIONS DONE
7357 045722                          305$:
7358
7359 045722                                  EXIT    TST                     ;////////// EXIT TEST //////////
     045722  104432                                                                                  TRAP    C$EXIT
     045724  C02112                                                                                  .WORD   L10056-.
7360
7361
7362                                 ;+
7363                                 ;LOCAL STORAGE FOR THIS TEST
7364                                 ;-
7365
7366
7367 045726                          T17MSK:                                 ;MASK OF UNTESTED BITS IN READ STATUS BYTES
7368                                                                         ;UNTESTED BITS ARE SET TO 1
7369 045726     377                          .BYTE   ^C<000>                 ;BYTE 0 MASK
7370 045727     037                          .BYTE   ^C<340>                 ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
7371 045730     360                          .BYTE   ^C<017>                 ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
7372 045731     000                          .BYTE   0                       ;MAKE IT EVEN
7373
7374 045732                          T17EXP:                                 ;BEGIN EXPECTED DATA BUFFER
7375 045732  000000                          .WORD   0                       ;MESSAGE TYPE
7376 045734  000000                          .WORD   0                       ;DATA FIELD LENGTH
7377 045736  000000                          .WORD   0                       ;RBPCR
7378 045740  000000                          .WORD   0                       ;XST0
7379 045742  000000                          .WORD   0                       ;XST1
7380 045744  000000                          .WORD   0                       ;XST2
7381 045746  000000                          .WORD   0                       ;XST3
7382 045750  000000                          .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
7383 045752                          T17EXSTA: .BLKB 66.                     ;EXPECTED READ STATUS AND WRITE FIFO DATA
7384 046054                          T17EXEND:                                       ;END EXPECTED DATA BUFFER
7385
7386 046054                          T17WFDATA: .BLKB 66.                    ;WRITE FIFO EXPECTED DATA BUFFER
7387
7388                                 ;+
7389                                 ;LOCAL TEXT MESSAGES FOR TEST
7390                                 ;-
7391
7392 046156     106     111     106  TST17ID:        .ASCIZ  'FIFO Exerciser'
7393 046175     127     122     111  T17SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
7394 046232     127     122     111  T172SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
7395 046276     127     122     111  T173SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
7396 046343     127     122     111  T174SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
7397 046406     127     122     111  T175SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
```

```
7398 046452  127  122  111  T176SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
7399 046515  106  111  106  T171CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize'
7400 046577  122  145  141  T172CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
7401 046673  106  111  106  T173CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after WRITE FIFO'
7402 046757  106  111  106  T174CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO'
7403 047042  122  145  141  T175CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
7404 047116  106  111  106  T176CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO from an Empty FIFO'
7405 047224  106  111  106  T177CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after RESET FIFO'
7406                          .EVEN
7407
7408                         ;+
7409                         ; CLEAR MESSAGE BUFFER
7410                         ;-
7411 047310                  T17CLRBUF:                                    ;
7412 047310                          SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
7413 047314  012701  047572          MOV     #T17BFR,R1                    ;GET MESSAGE BUFFER ADDRESS
7414 047320  012702  000120          MOV     #T17BEND-T17BFR,R2            ;SIZE OF MESSAGE BUFFER IN BYTES
7415 047324  105021          10$:    CLRB    (R1)+                         ;CLEAR A BYTE
7416 047326  C05302                  DEC     R2                            ;DONE?
7417 047330  003375                  BGT     10$                           ;BR IF NO
7418 047332  000207                  RTS     PC                            ;RETURN
7419
7420                         ;+
7421                         ; SETUP T17PK2 PACKET FOR READ STATUS
7422                         ;-
7423 047334                  T17SRD:
7424 047334  004737  047310          JSR     PC,T17CLRBUF                  ;CLEAR MESSAGE BUFFER
7425 047340  012700  047730          MOV     #T17DT2,R0                    ;WRITE SUBSYSTEM DATA BUFFER
7426 047344  112720  000005          MOVB    #PW.RDSTATUS,(R0)+            ;STORE READ STATUS COMMAND IN BSEL0
7427 047350  105010                  CLRB    (R0)                          ;CLEAR BSEL1
7428 047352  000207                  RTS     PC                            ;RETURN
7429
7430                         ;+
7431                         ; SETUP T17PK2 PACKET FOR WRITE MISC RESET FIFO
7432                         ;-
7433 047354                  T17RSFIF:
7434 047354  004737  047310          JSR     PC,T17CLRBUF                  ;CLEAR MESSAGE BUFFER
7435 047360  012700  047730          MOV     #T17DT2,R0                    ;WRITE SUBSYSTEM DATA BUFFER
7436 047364  112720  000010          MOVB    #PW.WMISC,(R0)+               ;STORE WRITE MISCELLANEOUS IN BSEL0
7437 047370  112710  000030          MOVB    #MS.RSFIF!MS.RSTAP,(R0)       ;STORE BSEL1 CL AR FIFO CODES
7438 047374  000207                  RTS     PC                            ;RETURN
7439
7440                         ;+
7441                         ; SETUP T17PK2 PACKET FOR WRITE NPR
7442                         ;
7443                         ; INPUT:
7444                         ;       R0 CONTAINS BSEL1 NPR DATA
7445                         ;
7446                         ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
7447                         ;-
7448 047376                  T17SNPR:
7449 047376  004737  047310          JSR     PC,T17CLRBUF                  ;CLEAR MESSAGE BUFFER
7450 047402  012701  047730          MOV     #T17DT2,R1                    ;WRITE SUBSYSTEM DATA BUFFER
7451 047406  112721  000011          MOVB    #PW.WNPR,(R1)+                ;STORE WRITE NPR IN BSEL0
7452 047412  052700  000020          BIS     #NP.WRP,R0                    ;DON'T WRITE WRONG PARITY
7453 047416  110011                  MOVB    R0,(R1)                       ;STORE NPR DATA IN BSEL1
7454 047420  000207                  RTS     PC                            ;RETURN
```

```
7455
7456                                    ;+
7457                                    ; SETUP T17PK2 PACKET FOR WRITE FIFO
7458                                    ;
7459                                    ; INPUT:
7460                                    ;       R0 CONTAINS BYTE COUNT
7461                                    ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
7462                                    ;-
7463 047422                            T17WFIF:
7464 047422                                    SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
7465 047426  004737  047310                    JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
7466 047432  012702  047730                    MOV     #T17DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
7467 047436  112722  000004                    MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
7468 047442  110022                            MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
7469 047444  005022                            CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
7470 047446  112122                    10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
7471 047450  005300                            DEC     R0                      ;DONE ALL BYTES?
7472 047452  003375                            BGT     10$                     ;BR IF NO
7473 047454  000207                            RTS     PC                      ;RETURN
7474
7475                                    ;+
7476                                    ; SETUP T17PK2 PACKET FOR READ FIFO
7477                                    ;
7478                                    ; INPUT:
7479                                    ;       R0 CONTAINS SEL2 BYTE COUNT
7480                                    ;-
7481 047456                            T17RFIF:
7482 047456  004737  047310                    JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
7483 047462  012701  047730                    MOV     #T17DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
7484 047466  112721  000003                    MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
7485 047472  110021                            MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
7486 047474  000207                            RTS     PC                      ;RETURN
7487                                    ;+
7488                                    ; CLEAR EXPECTED DATA MESSAGE BUFFER
7489                                    ;-
7490 047476                            T17CLEXP:
7491 047476  012701  045732                    MOV     #T17EXP,R1              ;GET EXPD ADDRESS
7492 047502  012700  000122                    MOV     #T17EXEND-T17EXP,R0     ;GET EXPD SIZE
7493 047506  105021                    10$:    CLRB    (R1)+                   ;CLEAR A BYTE
7494 047510  005300                            DEC     R0                      ;DONE?
7495 047512  003375                            BGT     10$                     ;BR IF NO
7496 047514  000207                            RTS     PC                      ;RETURN
7497
7498                                    ;+
7499                                    ;Set WORDS 0-7 of expd message buffer = to recv since not testing
7500                                    ;-
7501 047516                            T17SETEXP:
7502 047516  012702  045732                    MOV     #T17EXP,R2              ;GET EXPD
7503 047522  012703  047572                    MOV     #T17BFR,R3              ;GET READ STATUS RECV BUFFER
7504 047526  012700  000010                    MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
7505 047532  012322                    5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
7506 047534  005300                            DEC     R0                      ;DONE WORDS 0-7 WORDS?
7507 047536  003375                            BGT     5$                      ;BR IF NO
7508 047540  000207                            RTS     PC                      ;RETURN
7509
7511         047550                            .=<.+10>&177770
7513                                    ;
```

```
                                ;WRITE CHARACTERISTICS COMMAND PACKET
                                ;
047550                          T17PACKET:                              ;COMMAND PACKET FOR TEST
047550  100004                          .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
047552  047560                          .WORD   T17DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
047554  000000                          .WORD   0
047556  000012                          .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE

047560                          T17DATA:                                ;CHARACTERISTICS DATA BLOCK
047560  047572                          .WORD   T17BFR                  ;ADDRESS OF MESSAGE BUFFER
047562  000000                          .WORD   0
047564  000024                          .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
047566  000000                          .WORD   0                       ;ESS,ENB,EAI,ERI
047570  000000                          .WORD   0                       ;EXTENDED FEATURES UNIT NO. ETC.


                                ;MESSAGE BUFFER FOR ALL TEST 6 COMMANDS

047572                          T17BFR:                                 ;BEGIN MESSAGE BUFFER
047572  000000                          .WORD   0                       ;MESSAGE TYPE
047574  000000                          .WORD   0                       ;DATA FIELD LENGTH
047576  000000                          .WORD   0                       ;RBPCR
047600  000000                          .WORD   0                       ;XST0
047602  000000                          .WORD   0                       ;XST1
047604  000000                          .WORD   0                       ;XST2
047606  000000                          .WORD   0                       ;XST3
047610  000000                          .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
047612                          T17BFSTA: .BLKB 64.                     ;READ STATUS AND WRITE FIFO BUFFER
047712                          T17BEND:                                ;END OF MESSAGE BUFFER
                                ;
                                ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
                                ;
                                        .=<.+10>&177770
047720                          T17PK2:
047720  100006                          .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
047722  047730                          .WORD   T17DT2                  ;LOW ADDRESS OF DATA BLOCK
047724  000000                          .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
047726  000012                          .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE

047730                          T17DT2:                                 ;DATA BLOCK
047730     000                          .BYTE   0                       ;BSEL0
047731     000                          .BYTE   0                       ;BSEL1
047732  000000                          .WORD   0                       ;SEL2
047734                                  .BLKB   66.                     ;WRITE FIFO DATA OUTPUT BUFFER

                                ENDTST
050036                                                                  L10056:
050036                                                                          TRAP    C$ETST
050036  104401

                                        .SBTTL  TEST  7: STATIC TRANSPORT BUS INTERFACE TEST
                                ;+
                                ; TEST DESCRIPTION:
                                ;
                                ;
                                ; TEST STEPS:
                                ;
                                ;    REPEAT FOR LOOPCNT
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
                                        ;       BEGIN
                                        ;       Write to TSSR register to soft initialize the controller
                                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                                        ;       If Extended Features Hardware Switch Clear then:
                                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                                        ;       Do WRITE CHARACTERISTICS to select reserved unit 7
                                        ;       Do a Write Subsystem READ STATUS
                                        ;       If any transport interface signals are asserted then Print Error
                                        ;       END
                                        ;--

                                                BGNTST
050040                                                                                  T7::
050040
050040  012700  050546                           MOV     #TST18ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
050044  004737  016510                           JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
050050  012737  000012  002216                   MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
050056                                  T18LOOP:
                                        ;       Write to TSSR register to soft initialize the controller
050056                                  5$:
050056  004737  015774                           JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
050062  103405                                   BCS     10$                     ;BR IF SOFT INIT OKAY
050064  010001                                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
050066                                           ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
050066  104455                                                                                  TRAP    C$ERDF
050070  001274                                                                                  .WORD   700
050072  003652                                                                                  .WORD   SFIERR
050074  012034                                                                                  .WORD   SFIMSG

                                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
050076  005037  002222                   10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
050102  012704  051230                           MOV     #T18PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
050106  004737  010662                           JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
050112                                           FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
050126  103407                                   BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
050130  010004                                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
050132                                           NEXT.ERRNO
050132                                  12$:     ERRDF   ERRNO,T18SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
050132  104455                                                                                  TRAP    C$ERDF
050134  001275                                                                                  .WORD   701
050136  050605                                                                                  .WORD   T18SSR
050140  012046                                                                                  .WORD   PKTSSR
050142  005237  002222                           INC     FATFLG                  ;SET FATAL ERROR FLAG
050146                                  15$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
050146  104406                                                                                  TRAP    C$CLP1

                                        ;       If Extended Features Hardware Switch Clear then:
                                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
050150  012701  051252                           MOV     #T18BFR,R1              ;MESSAGE BUFFER ADDRESS
050154  032761  000200  000012                   BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
050162  001026                                   BNE     30$                     ;BR IF YES
050164  004737  051076                           JSR     PC,T18SMISC             ;SETUP PACKET FOR WRITE MISCELLANEOUS
050170  012704  051300                           MOV     #T18PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
050174  010465  000000                           MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
050200  004737  016336                           JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
050204                                           FORCERROR       22$                     ;@@FORCE ERROR IF FORCER=1
050220  103407                                   BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
 7622 050222  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7623 050224                                NEXT.ERRNO
 7624 050224                        22$:    ERRDF   ERRNO,T182SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      050224  104455                                                                TRAP    C$ERDF
      050226  001276                                                                .WORD   702
      050230  050642                                                                .WORD   T182SSR
      050232  012046                                                                .WORD   PKTSSR
 7625 050234  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 7626 050240                        30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      050240  104406                                                                TRAP    C$CLP1
 7627
 7628
 7629                               ;       Do WRITE CHARACTERISTICS to select reserved unit 7
 7630 050242  005037  002222                CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 7631 050246  012704  051230                MOV     #T18PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 7632 050252  004737  010662                JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 7633 050256                                FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
 7634 050272  103407                        BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 7635 050274  C10001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7636 050276                                NEXT.ERRNO
 7637 050276                        42$:    ERRDF   ERRNO,T18SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      050276  104455                                                                TRAP    C$ERDF
      050300  001277                                                                .WORD   703
      050302  050605                                                                .WORD   T18SSR
      050304  012046                                                                .WORD   PKTSSR
 7638 050306  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 7639 050312                        50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      050312  104406                                                                TRAP    C$CLP1
 7640
 7641                               ;       Clear message buffer
 7642 050314  012701  051252                MOV     #T18BFR,R1              ;GET MESSAGE BUFFER ADDRESS
 7643 050320  013700  051244                MOV     T18DATA+4,R0            ;SIZE OF MESSAGE BUFFER IN BYTES
 7644 050324  105021                60$:    CLRB    (R1)+                   ;CLEAR A BYTE
 7645 050326  005300                        DEC     R0                      ;DONE?
 7646 050330  003375                        BGT     60$                     ;BR IF NO
 7647                               ;       Do a Write Subsystem READ STATUS
 7648 050332  004737  051056                JSR     PC,T18SRD               ;SETUP PACKET FOR READ STATUS
 7649 050336  012704  051300                MOV     #T18PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7650 050342  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7651 050346  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7652 050352                                FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
 7653 050366  103407                        BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
 7654 050370  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7655 050372                                NEXT.ERRNO
 7656 050372                        62$:    ERRDF   ERRNO,T183SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      050372  104455                                                                TRAP    C$ERDF
      050374  001300                                                                .WORD   704
      050376  050706                                                                .WORD   T183SSR
      050400  012046                                                                .WORD   PKTSSR
 7657 050402  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 7658 050406                        70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      050406  104406                                                                TRAP    C$CLP1
 7659
 7660
 7661                               ;       Set first 8 words of expd message buffer = to recv since not testing
 7662                               ;       Set unused bits in Read Status expd equal rcvd
 7663 050410  004737  051120                JSR     PC,T18SETEXP            ;SET SOME EXPD TO RECV
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
7664                                    ;       If any transport interface signals are asserted then Print Error
7665 050414  005000                             CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
7666 050416  012701  051252                     MOV     #T18BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
7667 050422  012702  050516                     MOV     #T18EXP,R2              ;EXPD ADDRESS
7668 050426  012703  000012                     MOV     #10.,R3                 ;NUMBER OF WORDS TO COMPARE
7669 050432  004737  011500                     JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
7670 050436                                     FORCERROR       82$,NOTSSR      ;BAD
7671 050446  103404                             BCS     90$                     ;BR IF YES
7672 050450                                     NEXT.ERRNO
7673 050450                             82$:    ERRHRD  ERRNO,T18CMP,MSGSTAT    ;REPORT ERROR
     050450  104456                                                                     TRAP    C$ERHRD
     050452  001301                                                                     .WORD   705
     050454  050753                                                                     .WORD   T18CMP
     050456  012350                                                                     .WORD   MSGSTAT
7674 050460                             90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     050460  104406                                                                     TRAP    C$CLP1
7675
7676 050462  005737  002222                     TST     FATFLG                  ;ANY FATAL ERRORS ?
7677 050466  001402                             BEQ     160$                    ;BRANCH IF NOT
7678 050470  004737  017202                     JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
7679 050474  004737  016456             160$:   JSR     PC,TSTLOOP              ;DO ITERATIONS?
7680 050500  103002                             BCC     165$                    ;BR IF NO
7681 050502  000137  050056                     JMP     T18LOOP                 ;LOOP UNTIL ITERATIONS DONE
7682 050506                             165$:
7683 050506                                     EXIT    TST
     050506  104432                                                                     TRAP    C$EXIT
     050510  000606                                                                     .WORD   L10065-.
7684
7685
7686                                    ;+
7687                                    ;LOCAL STORAGE FOR THIS TEST
7688                                    ;-
7689
7690 050512                             T18MSK:                                 ;MASK OF UNUSED BITS IN READ STATUS BYTES
7691 050512     377                             .BYTE   ^C<000>                 ;BYTE 0 MASK
7692 050513     037                             .BYTE   ^C<340>                 ;BYTE 1
7693 050514     100                             .BYTE   ^C<277>                 ;BYTE 2
7694 050515     000                             .BYTE   0                       ;MAKE IT EVEN
7695
7696 050516                             T18EXP:                                 ;EXPECTED DATA BUFFER
7697 050516  000000                             .WORD   0                       ;MESSAGE TYPE
7698 050520  000000                             .WORD   0                       ;DATA FIELD LENGTH
7699 050522  000000                             .WORD   0                       ;RBPCR
7700 050524  000000                             .WORD   0                       ;XST0
7701 050526  000000                             .WORD   0                       ;XST1
7702 050530  000000                             .WORD   0                       ;XST2
7703 050532  000000                             .WORD   0                       ;XST3
7704 050534  000000                             .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUB)
7705 050536  000000                             .WORD   0                       ;READ STATUS BYTE 1/0
7706 050540  000000                             .WORD   0                       ;READ STATUS BYTE 2
7707
7708 050542     377     020             T18XS:  .BYTE   377,020                 ;READ STATUS BYTE 0/1 EXPECTED BASE
7709 050544  000000                             .WORD   0                       ;READ STATUS BYTE 2 EXPECTED BASE
7710
7711                                    ;+
7712                                    ;LOCAL TEXT MESSAGES FOR TEST
7713                                    ;-
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
7714
7715 050546     123     164     141  TST18ID:        .ASCIZ  'Static Transport Bus Interface'
7716 050605     127     122     111  T18SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
7717 050642     127     122     111  T182SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
7718 050706     127     122     111  T183SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
7719 050753     124     162     141  T18CMP: .ASCIZ  'Transport Bus Interface Signals NOT Negated After Unit 7 Selected'
7720                                                 .EVEN
7721
7722                                 ;+
7723                                 ; SETUP T18PK2 PACKET FOR READ STATUS
7724                                 ;-
7725 051056                         T18SRD:
7726 051056                                 SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
7727 051062  012700  051310                MOV     #T18DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
7728 051066  112720  000005                MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
7729 051072  105010                        CLRB    (R0)                    ;CLEAR BSEL1
7730 051074  000207                        RTS     PC                      ;RETURN
7731
7732                                 ;+
7733                                 ; SETUP T18PK2 PACKET FOR WRITE MISC.
7734                                 ;-
7735 051076                         T18SMISC:
7736 051076                                 SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
7737 051102  012700  051310                MOV     #T18DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
7738 051106  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
7739 051112  112710  000200                MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
7740 051116  000207                        RTS     PC                      ;RETURN
7741
7742                                 ;+
7743                                 ;Set first 8 words of expd message buffer = to recv since not testing
7744                                 ;       Set unused bits in Read Status expd equal rcvd
7745                                 ;-
7746 051120                         T18SETEXP:
7747 051120  012702  050516                MOV     #T18EXP,R2              ;GET EXPD
7748 051124  012703  051252                MOV     #T18BFR,R3              ;GET READ STATUS RECV BUFFER
7749 051130  012700  000010                MOV     #8.,R0                  ;SET FIRST 8 WORDS EXP=RECV
7750 051134  012322                5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
7751 051136  005300                        DEC     R0                      ;DONE FIRST 8 WORDS?
7752 051140  003375                        BGT     5$                      ;BR IF NO
7753 051142  012701  050512                MOV     #T18MSK,R1              ;GET UNUSED BIT MASK
7754 051146  013712  050542                MOV     T18XS,(R2)              ;SETUP BASE EXPECTED BYTE 1/0
7755 051152  013762  050544  000002        MOV     T18XS+2,2(R2)           ;SETUP BASE EXPECTED BYTE 2
7756 051160  011300                        MOV     (R3),R0                 ;GET RECV BYTE 1 AND BYTE 0
7757 051162  041100                        BIC     (R1),R0                 ;CLEAR ALL BUT UNUSED
7758 051164  040012                        BIC     R0,(R2)                 ;CLEAR UNUSED IN EXP
7759 051166  050012                        BIS     R0,(R2)                 ;SET UNUSED EXPD=RECV FOR COMPARE
7760 051170  016300  000002                MOV     2(R3),R0                ;GET RECV BYTE 2
7761 051174  046100  000002                BIC     2(R1),R0                ;CLEAR ALL BUT UNUSED
7762 051200  040062  000002                BIC     R0,2(R2)                ;CLEAR UNUSED IN EXPD
7763 051204  050062  000002                BIS     R0,2(R2)                ;SET UNUSED EXPD=RECV FOR COMPARE
7764 051210  105062  000003                CLRB    3(R2)                   ;CLEAR EXPD BYTE 3 (UNUSED)
7765 051214  105063  000003                CLRB    3(R3)                   ;CLEAR RECV BYTE 3 (UNUSED)
7766 051220  000207                        RTS     PC                      ;RETURN
7767
7769         051230                        .=<.+10>&177770
7771                                 ;
7772                                 ;WRITE CHARARTERISTICS COMMAND PACKET
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
7773
7774 051230                             ;
7775 051230  100004                     T18PACKET:                              ;COMMAND PACKET FOR TEST
7776 051232  051240                             .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
7777 051234  000000                             .WORD   T18DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
7778 051236  000012                             .WORD   0
7779                                            .WORD   10.                     ;MESSAGE PACKET MINIMUM SIZE
7780 051240                             T18DATA:                                ;CHARACTERISTICS DATA BLOCK
7781 051240  051252                             .WORD   T18BFR                  ;ADDRESS OF MESSAGE BUFFER
7782 051242  000000                             .WORD   0
7783 051244  000024                             .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
7784 051246  000000                             .WORD   0                       ;ESS,ENB,EAI,ERI
7785 051250  000007                             .WORD   7                       ;SELECT RESERVED UNIT 7
7786
7787
7788 051252                             T18BFR:                                 ;MESSAGE BUFFER
7789 051252  000000                             .WORD   0                       ;MESSAGE TYPE
7790 051254  000000                             .WORD   0                       ;DATA FIELD LENGTH
7791 051256  C00000                             .WORD   0                       ;RBPCR
7792 051260  000000                             .WORD   0                       ;XST0
7793 051262  000000                             .WORD   0                       ;XST1
7794 051264  000000                             .WORD   0                       ;XST2
7795 051266  000000                             .WORD   0                       ;XST3
7796 051270  000000                             .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
7797 051272  000000                             .WORD   0                       ;READ STATUS BYTE 1/O RETURNED
7798 051274  000000                             .WORD   0                       ;READ STATUS BYTE 2
7799                                    ;
7800                                    ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
7801                                    ;
7803         051300                             .=<.+10>&177770
7805 051300                             T18PK2:
7806 051300  100006                             .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
7807 051302  051310                             .WORD   T18DT2                  ;LOW ADDRESS OF DATA BLOCK
7808 051304  000000                             .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
7809 051306  000010                             .WORD   8.                      ;BUFFER EXTENT
7810
7811 051310                             T18DT2:                                 ;DATA BLOCK
7812 051310     000                             .BYTE   0                       ;BSEL0
7813 051311     000                             .BYTE   0                       ;BSEL1
7814 051312  000000                             .WORD   0                       ;SEL2
7815 051314  000000                             .WORD   0                       ;DATA
7816
7817
7818 051316                             ENDTST
     051316                                                             L10065:
     051316  104401                                                             TRAP    L$ETST
7819                                            .SBTTL  TEST  8: TRANSPORT BUS INTERFACE LOOPBACK TEST
7820                                    ;**
7821                                    ; TEST DESCRIPTION:
7822                                    ;
7823                                    ;       This test verifies the controller's Transport Bus
7824                                    ;       drivers, receivers, and signal loopback logic. Note
7825                                    ;       that the Static Transport Bus test must have run
7826                                    ;       correctly for this test to provide meaningful results.
7827                                    ;
7828                                    ; TEST STEPS:
7829                                    ;
```

TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST

```
                                        ;  REPEAT FOR LOOPCNT
                                        ;    BEGIN
                                        ;       Do Subtest 1    - Loopback Control signals test
                                        ;       Do Subtest 2    - Loopback Read/Write signals test
                                        ;       Do Subtest 3    - Loopback Write Strobe test
                                        ;       Do Subtest 4    - Loopback Read Strobe test
                                        ;    END
                                        ;--


051320                                          BGNTST
051320                                                                          T8::
051320  012700  057532                          MOV     #TST19ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
051324  004737  016510                          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
051330  012737  000012  002216                  MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
051336                                  T19LOOP:

                                                .SBTTL  TEST  8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST


                                        ;++
                                        ; TEST 8: SUBTEST 1:
                                        ;
                                        ; SUBTEST DESCRIPTION:
                                        ;
                                        ;       This subtest verifies the Transport Control loopback
                                        ;       path can transmit and receive correctly. The
                                        ;       control signals are all loopback signals other
                                        ;       than the read/write data (IW<7:0> and IR<7:0>).
                                        ;
                                        ; TEST STEPS:
                                        ;
                                        ;       The loopback signals IFAD,ITAD0,ITAD1 are the tape unit select
                                        ;       lines. Since reserved unit 7 must remain selected these signals
                                        ;       are always set low. This further means the signals they drive
                                        ;       (ISPEED,IRDY,IONL) are only tested in the low state.
                                        ;
                                        ; BEGIN
                                        ;       Write to TSSR register to soft initialize the controller
                                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                                        ;       If Extended Features Hardware Switch Clear then:
                                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                                        ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
                                        ;       Do a Write Subsystem WRITE MPR to set tape direction out and Loopback
                                        ;       Do Write Subsystem Write Control to CLEAR loopback signals group 1.
                                        ;       Do Write Subsystem Write Format to CLEAR loopback signals group 2.
                                        ;        (the loopback signals have to be cleared here due to the flip-flops
                                        ;         that are set on a 1 to 0 transition (IHER,IFMK,ICER))
                                        ;       Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
                                        ;       Do a Write Subsystem READ STATUS
                                        ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
                                        ;         signals NOT=0 Then Print Error
                                        ;
                                        ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
                                        ;    BEGIN
                                        ;       Do Write Subsystem Write Control to Drive  oopback signals group 1.
                                        ;       Do Write Subsystem Write Format to Drive .oopback signals group 2.
                                        ;       Do a Write Subsystem READ STATUS
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
                                ;       If loopback data NOT= data sent Then Print Error
                                ;       Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
                                ;       Do a Write Subsystem READ STATUS
                                ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
                                ;         signals NOT=0 Then Print Error
                                ;    END
                                ;--
051336                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
051336                                                                                    T8.1:
051336  104402                                                                            TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
051340                          5$:
051340  004737  015774                  JSR     PC,SOFT INIT            ;WRITE TO TSSR TO SOFT INITIALIZE
051344  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
051346  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
051350                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
051350  104455                                                                            TRAP    C$ERDF
051352  001440                                                                          .WORD   800
051354  003652                                                                          .WORD   SFIERR
051356  012034                                                                          .WORD   SFIMSG

                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
051360  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
051364  012704  061660                  MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
051370  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
051374                                  FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
051410  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
051412  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
051414                                  NEXT.ERRNO
051414                          12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
051414  104455                                                                            TRAP    C$ERDF
051416  001441                                                                          .WORD   801
051420  057573                                                                          .WORD   T19SSR
051422  012046                                                                          .WORD   PKTSSR
051424  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
051430                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
051430  104406                                                                            TRAP    C$CLP1
                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
051432  012701  061702                  MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
051436  032761  000200  000012          BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
051444  001026                          BNE     30$                     ;BR IF YES
051446  004737  061532                  JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
051452  012704  062030                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
051456  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
051462  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
051466                                  FORCERROR       22$                     ;@@FORCE ERROR IF FORCER=1
051502  103407                          BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
051504  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
051506                                  NEXT.ERRNO
051506                          22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
051506  104455                                                                            TRAP    C$ERDF
051510  001442                                                                          .WORD   802
051512  057630                                                                          .WORD   T192SSR
051514  012046                                                                          .WORD   PKTSSR
051516  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
051522                          30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
      051522  104406                                                                TRAP    C$CLP1
 7932
 7933 051524  005037  002222          ;       Do WRITE CHARACTERISTICS to select reserved unit 7
                                      CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 7934 051530  012704  061660                  MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 7935 051534  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 7936 051540                                  FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
 7937 051554  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 7938 051556  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7939 051560                                  NEXT.ERRNO
 7940 051560                          42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      051560  104455                                                                TRAP    C$ERDF
      051562  001443                                                                .WORD   803
      051564  057573                                                                .WORD   T19SSR
      051566  012046                                                                .WORD   PKTSSR
 7941 051570  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 7942 051574                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      051574  104406                                                                TRAP    C$CLP1
 7943                                 ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
 7944 051576  C12700  000100                  MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
 7945 051602  052700  000040                  BIS     #NP.LOOP,R0             ;SET LOOPBACK ENABLE
 7946 051606  004737  061372                  JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
 7947 051612  012704  062030                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7948 051616  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7949 051622  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7950 051626                                  FORCERROR       62$                     ;@@FORCE ERROR IF FORCER=1
 7951 051642  103407                          BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
 7952 051644  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7953 051646                                  NEXT.ERRNO
 7954 051646                          62$:    ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      051646  104455                                                                TRAP    C$ERDF
      051650  001444                                                                .WORD   804
      051652  057741                                                                .WORD   T194SSR
      051654  012046                                                                .WORD   PKTSSR
 7955 051656  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 7956 051662                          70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      051662  104406                                                                TRAP    C$CLP1
 7957                                 ;       Do Write Subsystem Write Control to CLEAR loopback signals group 1.
 7958                                 ;       Do Write Subsystem Write Format to CLEAR loopback signals group 2.
 7959                                 ;        (the loopback signals have to be cleared here due to the flip-flops
 7960                                 ;         that are set on a 1 to 0 transition (IHER,IFMK,ICER))
 7961 051664  005000                          CLR     R0                      ;WRITE 0'S
 7962 051666  042700  000200                  BIC     #WC.IFAD,R0             ;IFAD MUST  ALWAYS =0
 7963 051672  042700  000100                  BIC     #WC.IOTAD,R0            ;ITADO MUST ALWAYS =0
 7964 051676  042700  000040                  BIC     #WC.I1TAD,R0            ;ITAD1 MUST ALWAYS =0
 7965 051702  004737  061472                  JSR     PC,T19WCTL              ;SETUP PACKET FOR WRITE CONTROL
 7966 051706  012704  062030                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7967 051712  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7968 051716  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7969 051722                                  FORCERROR       82$                     ;@@FORCE ERROR IF FORCER=1
 7970 051736  103407                          BCS     90$                     ;BR IF CARRY SET (GOOD RETURN)
 7971 051740  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7972 051742                                  NEXT.ERRNO
 7973 051742                          82$:    ERRDF   ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      051742  104455                                                                TRAP    C$ERDF
      051744  001445                                                                .WORD   805
      051746  060113                                                                .WORD   T197SSR
      051750  012046                                                                .WORD   PKTSSR
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
 7974 051752  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 7975 051756                         90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      051756  104406                                                                TRAP    C$CLP1
 7976 051760  005000                         CLR     R0                      ;SET FORMAT DRIVE DATA=0
 7977 051762  004737  061512                 JSR     PC,T19WFMT              ;SETUP PACKET FOR WRITE FORMAT
 7978 051766  012704  062030                 MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7979 051772  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7980 051776  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7981 052002                                 FORCERROR       102$                    ;@@DFORCE ERROR IF FORCEP=1
 7982 052016  103407                         BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
 7983 052020  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7984 052022                                 NEXT.ERRNO
 7985 052022                         102$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052022  104455                                                                TRAP    C$ERDF
      052024  001446                                                               .WORD   806
      052026  060162                                                               .WORD   T198SSR
      052030  012046                                                               .WORD   PKTSSR
 7986 052032  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 7987 052036                         110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052036  104406                                                                TRAP    C$CLP1
 7988                                ;       Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
 7989 052040  004737  061350                 JSR     PC,T19RSFIF             ;SETUP PKT FOR WRITE MISC Reset Tape Status F-FLOPS
 7990 052044  012704  062030                 MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7991 052050  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7992 052054  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7993 052060                                 FORCERROR       122$                    ;@@DFORCE ERROR IF FORCER=1
 7994 052074  103407                         BCS     130$                    ;BR IF CARRY SET (GOOD RETURN)
 7995 052076  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7996 052100                                 NEXT.ERRNO
 7997 052100                         122$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052100  104455                                                                TRAP    C$ERDF
      052102  001447                                                               .WORD   807
      052104  057630                                                               .WORD   T192SSR
      052106  012046                                                               .WORD   PKTSSR
 7998 052110  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 7999 052114                         130$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052114  104406                                                                TRAP    C$CLP1
 8000                                ;       Do a Write Subsystem READ STATUS
 8001                                ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
 8002                                ;         signals NOT=0 Then Print Error
 8003 052116  004737  061330                 JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
 8004 052122  012704  062030                 MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8005 052126  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8006 052132  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8007 052136                                 FORCERROR       132$                    ;@@DFORCE ERROR IF FORCER=1
 8008 052152  103407                         BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
 8009 052154  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8010 052156                                 NEXT.ERRNO
 8011 052156                         132$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052156  104455                                                                TRAP    C$ERDF
      052160  001450                                                               .WORD   808
      052162  057674                                                               .WORD   T193SSR
      052164  012046                                                               .WORD   PKTSSR
 8012 052166  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 8013 052172                         140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052172  104406                                                                TRAP    C$CLP1
 8014 052174  004737  061570                 JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
 8015 052200  012701  057432                    MOV     @T19EXSTA,R1            ;GET EXPECTED READ STATUS
 8016 052204  012702  061722                    MOV     @T19BFSTA,R2            ;GET RECV READ STATUS
 8017 052210  011211                            MOV     (R2),(R1)               ;SET EXPD WORD @8 = RECV TEMP
 8018 052212  042711  002000                    BIC     @S1.ICER,(R1)           ;SET EXPD ICER  =0
 8019 052216  042711  001000                    BIC     @S1.IFMK,(R1)           ;SET EXPD IFMK  =0
 8020 052222  042711  000400                    BIC     @S1.IHER,(R1)           ;SET EXPD IHER  =0
 8021 052226  016261  000002  000002            MOV     2(R2),2(R1)             ;SET EXPD WORD @9 = RECV (NOT TESTING)
 8022 052234  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 8023 052236  012701  061702                    MOV     @T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8024 052242  012702  057412                    MOV     @T19EXP,R2              ;EXPD ADDRESS
 8025 052246  012703  000024                    MOV     @20.,R3                 ;NUMBER OF BYTES TO COMPARE
 8026 052252  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 8027 052256                                    FORCERROR       152$,NOTSSR     ;@@D
 8028 052266  103404                            BCS     160$                    ;BR IF YES
 8029 052270                                    NEXT.ERRNO
 8030 052270                            152$:   ERRHRD  ERRNO,T197CMP,MSGLOOP   ;REPORT ERROR
      052270  104456                                                                    TRAP    C$ERHRD
      052272  001451                                                                    .WORD   809
      052274  C60633                                                                    .WORD   T197CMP
      052276  013064                                                                    .WORD   MSGLOOP
 8031 052300                            160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052300  104406                                                                    TRAP    C$CLP1
 8032                                   ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
 8033 052302  005037  057344                    CLR     T19PREV                 ;INIT 1-0 TRANSITION FLAG
 8034 052306  012703  002752                    MOV     @TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
 8035 052312  012300                    200$:   MOV     (R3)+,R0                ;GET A TEST PATTERN
 8036 052314  010337  002316                    MOV     R3,TSTPTR               ;SAVE POINTER INTO TSTBLK
 8037 052320  042700  000200                    BIC     @WC.IFAD,R0             ;IFAD MUST  ALWAYS =0
 8038 052324  042700  000100                    BIC     @WC.IOTAD,R0            ;ITADO MUST ALWAYS =0
 8039 052330  042700  000040                    BIC     @WC.I1TAD,R0            ;ITAD1 MUST ALWAYS =0
 8040 052334  010037  002312                    MOV     R0,DATA                 ;SET DATA PATTERN
 8041                                   ;       Do Write Subsystem Write Control to Drive loopback signals group 1.
 8042                                   ;@@D    CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
 8043 052340  013700  002312                    MOV     DATA,R0                 ;GET TEST PATTERN
 8044 052344  004737  061614                    JSR     PC,T19CNVT              ;CONVERT PATTERN TO CONTROL DRIVE MASK
 8045                                                                           ;R0 CONTAINS WRITE CONTROL DATA HERE
 8046 052350  004737  061472                    JSR     PC,T19WCTL              ;SETUP PACKET FOR WRITE CONTROL
 8047 052354  012704  062030                    MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8048 052360  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8049 052364  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8050 052370                                    FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
 8051 052404  103407                            BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
 8052 052406  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8053 052410                                    NEXT.ERRNO
 8054 052410                            212$:   ERRDF   ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052410  104455                                                                    TRAP    C$ERDF
      052412  001452                                                                    .WORD   810
      052414  060113                                                                    .WORD   T197SSR
      052416  012046                                                                    .WORD   PKTSSR
 8055 052420  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
 8056 052424                            220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052424  104406                                                                    TRAP    C$CLP1
 8057
 8058                                   ;       Do Write Subsystem Write Format to Drive loopback signals group 2.
 8059                                   ;@@D    CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
 8060 052426  013700  002312                    MOV     DATA,R0                 ;GET TEST PATTERN
 8061 052432  004737  061614                    JSR     PC,T19CNVT              ;CONVERT PATTERN TO FORMAT DRIVE MASK
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
 8062 052436  000300                        SWAB    R0                      ;WRITE FORMAT DATA RETURNED IN HIGH BYTE
 8063 052440  004737  061512                JSR     PC,T19WFMT              ;SETUP PACKET FOR WRITE FORMAT
 8064 052444  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8065 052450  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8066 052454  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8067 052460                                FORCERROR       232$                    ;@@DFORCE ERROR IF FORCER=1
 8068 052474  103407                        BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
 8069 052476  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8070 052500                                NEXT.ERRNO
 8071 052500                        232$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052500  104455                                                                TRAP    C$ERDF
      052502  001453                                                                .WORD   811
      052504  060162                                                                .WORD   T198SSR
      052506  012046                                                                .WORD   PKTSSR
 8072 052510  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8073 052514                        240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052514  104406                                                                TRAP    C$CLP1
 8074                                ;       Do a Write Subsystem READ STATUS
 8075 052516  C04737  061330                JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
 8076 052522  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8077 052526  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8078 052532  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8079 052536                                FORCERROR       252$                    ;@@DFORCE ERROR IF FORCER=1
 8080 052552  103407                        BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
 8081 052554  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8082 052556                                NEXT.ERRNO
 8083 052556                        252$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052556  104455                                                                TRAP    C$ERDF
      052560  001454                                                                .WORD   812
      052562  057674                                                                .WORD   T193SSR
      052564  012046                                                                .WORD   PKTSSR
 8084 052566  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8085 052572                        260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052572  104406                                                                TRAP    C$CLP1
 8086                                ;       If loopback data NOT= data sent Then Print Error
 8087 052574  004737  061570                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 8088 052600  012701  057432                MOV     #T19EXSTA,R1             GET EXPECTED READ STATUS
 8089 052604  012702  061722                MOV     #T198FSTA,R2            ;GET RECV READ STATUS
 8090 052610  013711  002312                MOV     DATA,(R1)               ;SET EXPD WORD #8 TO TEST DATA FIRST
 8091 052614  013700  057344                MOV     T19PREV,R0              ;GET PREVIOUS DATA PATTERN
 8092 052620  013703  002312                MOV     DATA,R3                 ;GET CURRENT PATTERN
 8093 052624  012704  000400                MOV     #S1.IHER,R4             ;SETUP IHER EXPECTED
 8094 052630  040411                        BIC     R4,(R1)                 ;SET EXPD IHER  =0
 8095 052632  030400                        BIT     R4,R0                   ;PREVIOUS =1?
 8096 052634  001403                        BEQ     275$                    ;BR IF NO
 8097 052636  030403                        BIT     R4,R3                   ;CURRENT  =0?
 8098 052640  001001                        BNE     275$                    ;BR IF NO
 8099 052642  050411                        BIS     R4,(R1)                 ;SET EXPD IHER  =1
 8100 052644  012704  001000        275$:   MOV     #S1.IFMK,R4             ;SETUP IFMK EXPECTED
 8101 052650  040411                        BIC     R4,(R1)                 ;SET EXPD IFMK  =0
 8102 052652  030400                        BIT     R4,R0                   ;PREVIOUS =1?
 8103 052654  001403                        BEQ     280$                    ;BR IF NO
 8104 052656  030403                        BIT     R4,R3                   ;CURRENT  =0?
 8105 052660  001001                        BNE     280$                    ;BR IF NO
 8106 052662  050411                        BIS     R4,(R1)                 ;SET EXPD IFMK  =1
 8107 052664  012704  002000        280$:   MOV     #S1.ICER,R4             ;SETUP ICER EXPECTED
 8108 052670  040411                        BIC     R4,(R1)                 ;SET EXPD ICER  =0
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
 8109 052672  030400                          BIT     R4,R0                   ;PREVIOUS =1?
 8110 052674  001403                          BEQ     285$                    ;BR IF NO
 8111 052676  030403                          BIT     R4,R3                   ;CURRENT  =0?
 8112 052700  001001                          BNE     285$                    ;BR IF NO
 8113 052702  050411                          BIS     R4,(R1)                 ;SET EXPD ICER  =1
 8114 052704  011100                  285$:   MOV     (R1),R0                 ;GET EXPD WORD
 8115                                ;       If previous IIDENT=1 and current is IIDENT=1 then EXPD= 0 else 1
 8116 052706  012704  004000                  MOV     #S1.IIDENT,R4           ;IIDENT
 8117 052712  050400                          BIS     R4,R0                   ;ASSUME EXPD=1
 8118 052714  030437  057344                  BIT     R4,T19PREV              ;PREVIOUS IIDENT=1?
 8119 052720  001403                          BEQ     288$                    ;BR IF NO
 8120 052722  030403                          BIT     R4,R3                   ;IS CURRENT IIDENT=1?
 8121 052724  001401                          BEQ     288$                    ;BR IF NO
 8122 052726  040400                          BIC     R4,R0                   ;SET EXPD=0
 8123 052730  052700  040000          288$:   BIS     #S1.I2RES,R0            ;IRESV2 EXPD ALWAYS=1
 8124 052734  052700  020000                  BIS     #S1.I1RES,R0            ;IRESV1 EXPD ALWAYS=1
 8125 052740  042700  100000                  BIC     #S1.PARERR,R0           ;IGNORE PARERR
 8126 052744  032712  100000                  BIT     #S1.PARERR,(R2)         ;IS PARERR SET IN RECV?
 8127 052750  C01402                          BEQ     290$                    ;BR IF NO
 8128 052752  052700  100000                  BIS     #S1.PARERR,R0           ;SET IN EXPD
 8129 052756  010011                  290$:   MOV     R0,(R1)                 ;SETUP FINAL EXPD IN WORD #8
 8130 052760  016261  000002  000002          MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
 8131 052766  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 8132 052770  012701  061702                  MOV     #T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8133 052774  012702  057412                  MOV     #T19EXP,R2              ;EXPD ADDRESS
 8134 053000  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 8135 053004  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 8136 053010                                  FORCERROR       302$,NOTSSR     ;@@D
 8137 053020  103404                          BCS     310$                    ;BR IF YES
 8138 053022                                  NEXT.ERRNO
 8139 053022                          302$:   ERRHRD  ERRNO,T198CMP,MSGLOOP   ;REPORT ERROR
      053022  104456                                                          TRAP    C$ERHRD
      053024  001455                                                          .WORD   813
      053026  060721                                                          .WORD   T198CMP
      053030  013064                                                          .WORD   MSGLOOP
 8140 053032                          310$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      053032  104406                                                          TRAP    C$CLP1
 8141                                ;       Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
 8142 053034  004737  061350                  JSR     PC,T19RSFIF             ;SETUP PKT FOR WRITE MISC Reset STATUS
 8143 053040  012704  062030                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8144 053044  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8145 053050  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8146 053054                                  FORCERROR       322$            ;@@FORCE ERROR IF FORCER=1
 8147 053070  103407                          BCS     330$                    ;BR IF CARRY SET (GOOD RETURN)
 8148 053072  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8149 053074                                  NEXT.ERRNO
 8150 053074                          322$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      053074  104455                                                          TRAP    C$ERDF
      053076  001456                                                          .WORD   814
      053100  057630                                                          .WORD   T192SSR
      053102  012046                                                          .WORD   PKTSSR
 8151 053104  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 8152 053110                          330$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      053110  104406                                                          TRAP    C$CLP1
 8153                                ;       Do a Write Subsystem READ STATUS
 8154 053112  004737  061330                  JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
 8155 053116  012704  062030                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
8156 053122  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8157 053126  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8158 053132                                FORCERROR       342$            ;@@FORCE ERROR IF FORCER=1
8159 053146  103407                        BCS     350$                    ;BR IF CARRY SET (GOOD RETURN)
8160 053150  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8161 053152                                NEXT.ERRNO
8162 053152                        342$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053152  104455                                                                TRAP    C$ERDF
     053154  001457                                                                .WORD   815
     053156  057674                                                                .WORD   T193SSR
     053160  012046                                                                .WORD   PKTSSR
8163 053162  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8164 053166                        350$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053166  104406                                                                TRAP    C$CLP1
8165 053170  004737  061570                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
8166 053174  012701  057432                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
8167 053200  012702  061722                MOV     #T198FSTA,R2            ;GET RECV READ STATUS
8168 053204  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
8169 053206  042711  002000                BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
8170 053212  042711  001000                BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
8171 053216  042711  000400                BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
8172 053222  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
8173 053230  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
8174 053232  012701  061702                MOV     #T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
8175 053236  012702  057412                MOV     #T19EXP,R2              ;EXPD ADDRESS
8176 053242  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
8177 053246  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
8178 053252                                FORCERROR       362$,NOTSSR     ;@@D
8179 053262  103404                        BCS     370$                    ;BR IF YES
8180 053264                                NEXT.ERRNO
8181 053264                        362$:   ERRHRD  ERRNO,T197CMP,MSGSTAT   ;REPORT ERROR
     053264  104456                                                                TRAP    C$ERHRD
     053266  001460                                                                .WORD   816
     053270  060633                                                                .WORD   T197CMP
     053272  012350                                                                .WORD   MSGSTAT
8182 053274                        370$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053274  104406                                                                TRAP    C$CLP1
8183
8184 053276  013737  002312  057344        MOV     DATA,T19PREV            ;SETUP PREVIOUS DATA FOR EXPD CALC.
8185 053304  013703  002316                MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
8186 053310  020327  003062                CMP     R3,#TBLEND              ;END OF TSTBLK?
8187 053314  103002                        BHIS    400$                    ;BR IF YES
8188 053316  000137  052312                JMP     200$                    ;DO NEXT TSTBLK PATTERN
8189 053322                        400$:
8190
8191 053322                                ENDSUB                          ;////////////// END SUBTEST //////////
     053322                                                                        L10067:
     053322  104403                                                                TRAP    C$ESUB
8192
8193 053324  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
8194 053330  001402                        BEQ     460$                    ;BRANCH IF NOT
8195 053332  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
8196 053336                        460$:
8197
8198
8199
8200
```

```
                                        .SBTTL  TEST  8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST

                                ;++
                                ; TEST 8: SUBTEST 2:
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies the Read/Write data loopback path.
                                ;       The Read/Write data signals are IR<7:0> and IW<7:0>
                                ;       respectively.
                                ;
                                ; TEST STEPS:
                                ;
                                ;
                                ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
                                ;    BEGIN
                                ;       Write to TSSR register to soft initialize the controller
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
                                ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
                                ;       Do a WRITE NPR to set loopback and tape direction OUT
                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
                                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
                                ;       Do a WRITE NPR to set loopback and tape direction IN
                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
                                ;         to strobe loopback data into FIFO.
                                ;       Do a READ FIFO with tape direction IN to read data
                                ;       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
                                ;--
053336                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
053336                                                                                  T8.2:
053336  104402                                                                          TRAP    C$BSUB
                                ;       Write to TSSR register to soft initialize the controller
053340                          5$:
053340  004737  015774                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
053344  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
053346  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
053350                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
053350  104455                                                                          TRAP    C$ERDF
053352  001460                                                                          .WORD   816
053354  003652                                                                          .WORD   SFIERR
053356  012034                                                                          .WORD   SFIMSG
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
053360  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
053364  012704  061660                  MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
053370  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
053374                                  FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
053410  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
053412  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
                                        NEXT.ERRNO
                              12$:      ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
053414  104455                                                                TRAP    C$ERDF
053416  001461                                                                .WORD   817
053420  057573                                                                .WORD   T19SSR
053422  012046                                                                .WORD   PKTSSR
053424  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
053430                        15$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
053430  104406                                                                TRAP    C$CLP1
                              ;         If Extended Features Hardware Switch Clear then:
                              ;           Do Write Subsystem Write Miscellaneous to Set Extended Features.
053432  012701  061702                MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
053436  032761  000200  000012        BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
053444  001026                        BNE     30$                     ;BR IF YES
053446  004737  061532                JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
053452  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
053456  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
053462  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
053466                                FORCERROR       22$                     ;@@FORCE ERROR IF FORCER=1
053502  103407                        BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
053504  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
053506                                NEXT.ERRNO
053506                        22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
053506  104455                                                                TRAP    C$ERDF
053510  001462                                                                .WORD   818
053512  057630                                                                .WORD   T192SSR
053514  012046                                                                .WORD   PKTSSR
053516  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
053522                        30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
053522  104406                                                                TRAP    C$CLP1
                              ;         Do WRITE CHARACTERISTICS to select reserved unit 7
053524  012704  061660                MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
053530  004737  010662                JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
053534                                FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
053550  103407                        BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
053552  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
053554                                NEXT.ERRNO
053554                        42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
053554  104455                                                                TRAP    C$ERDF
053556  001463                                                                .WORD   819
053560  057573                                                                .WORD   T19SSR
053562  012046                                                                .WORD   PKTSSR
053564  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
053570                        50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
053570  104406                                                                TRAP    C$CLP1


                              ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
053572  012703  002752                MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
053576  012337  002312        100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
053602  042737  177400  002312        BIC     #^C<377>,DATA           ;DATA IS BYTE
053610  010337  002316                MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
                              ;         Do a WRITE NPR to set loopback and tape direction OUT
053614  012700  000100                MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
053620  052700  000040                BIS     #NP.LOOP,R0             ;SET LOOPBACK
053624  004737  061372                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
053630  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
```

TEST 8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST

```
8294 053634  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8295 053640  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8296 053644                                FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
8297 053660  103407                        BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
8298 053662  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8299 053664                                NEXT.ERRNO
8300 053664                        102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053664  104455                                                                TRAP    C$ERDF
     053666  001464                                                                .WORD   820
     053670  057741                                                                .WORD   T194SSR
     053672  012046                                                                .WORD   PKTSSR
8301 053674  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8302 053700                        105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053700  104406                                                                TRAP    C$CLP1
8303                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8304 053702  012700  000001                MOV     #1,R0                   ;WRITE 1 BYTE
8305 053706  012701  002312                MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
8306 053712  004737  061436                JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
8307 053716  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8308 053722  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8309 053726  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8310 053732                                FORCERROR       107$                    ;@@DFORCE ERROR IF FORCER=1
8311 053746  103407                        BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
8312 053750  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8313 053752                                NEXT.ERRNO
8314 053752                        107$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053752  104455                                                                TRAP    C$ERDF
     053754  001465                                                                .WORD   821
     053756  060004                                                                .WORD   T195SSR
     053760  012046                                                                .WORD   PKTSSR
8315 053762  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8316 053766                        110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053766  104406                                                                TRAP    C$CLP1
8317                                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
8318 053770  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
8319 053774  004737  061416                JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
8320 054000  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8321 054004  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8322 054010  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8323 054014                                FORCERROR       122$                    ;@@DFORCE ERROR IF FORCER=1
8324 054030  103407                        BCS     130$                    ;BR IF CARRY SET (GOOD RETURN)
8325 054032  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8326 054034                                NEXT.ERRNO
8327 054034                        122$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054034  104455                                                                TRAP    C$ERDF
     054036  001466                                                                .WORD   822
     054040  060050                                                                .WORD   T196SSR
     054042  012046                                                                .WORD   PKTSSR
8328 054044  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8329 054050                        130$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054050  104406                                                                TRAP    C$CLP1
8330                                ;       Do a WRITE NPR to set loopback and tape direction IN
8331 054052  005000                        CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
8332 054054  052700  000040                BIS     #NP.LOOP,R0             ;SET LOOPBACK
8333 054060  004737  061372                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
8334 054064  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8335 054070  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

TEST 8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST

```
8336 054074  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8337 054100                                FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
8338 054114  103407                        BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
8339 054116  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8340 054120                                NEXT.ERRNO
8341 054120                        142$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054120  104455                                                                TRAP    C$ERDF
     054122  001467                                                                .WORD   823
     054124  057741                                                                .WORD   T194SSR
     054126  012046                                                                .WORD   PKTSSR
8342 054130  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8343 054134                        150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054134  104406                                                                TRAP    C$CLP1
8344                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
8345 054136  012700  000001                MOV     #1,R0                   ;WRITE 1 BYTE
8346 054142  012701  002312                MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
8347 054146  004737  061436                JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
8348 054152  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8349 054156  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8350 054162  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8351 054166                                FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
8352 054202  103407                        BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
8353 054204  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8354 054206                                NEXT.ERRNO
8355 054206                        162$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054206  104455                                                                TRAP    C$ERDF
     054210  001470                                                                .WORD   824
     054212  060004                                                                .WORD   T195SSR
     054214  012046                                                                .WORD   PKTSSR
8356 054216  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8357 054222                        170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054222  104406                                                                TRAP    C$CLP1
8358                                ;       Do a READ FIFO with tape direction IN to read data
8359                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
8360 054224  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
8361 054230  004737  061416                JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
8362 054234  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8363 054240  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8364 054244  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8365 054250                                FORCERROR       182$                    ;@@DFORCE ERROR IF FORCER=1
8366 054264  103407                        BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
8367 054266  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8368 054270                                NEXT.ERRNO
8369 054270                        182$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054270  104455                                                                TRAP    C$ERDF
     054272  001471                                                                .WORD   825
     054274  060050                                                                .WORD   T196SSR
     054276  012046                                                                .WORD   PKTSSR
8370 054300  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8371 054304                        190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054304  104406                                                                TRAP    C$CLP1
8372 054306  004737  061570                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
8373 054312  012701  057432                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
8374 054316  012702  061722                MOV     #T19RFSTA,R2            ;GET RECV READ STATUS
8375 054322  013711  002312                MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
8376 054326  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
8377 054334  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
```

TEST 8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST

```
 8378 054336  012701  061702                MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8379 054342  012702  057412                MOV     #T19EXP,R2              ;EXPD ADDRESS
 8380 054346  012703  000022                MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
 8381 054352  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 8382 054356                                FORCERROR       202$,NOTSSR     ;@@D
 8383 054366  103404                        BCS     210$                    ;BR IF YES
 8384 054370                                NEXT.ERRNO
 8385 054370                        202$:   ERRHRD  ERRNO,T199CMP,MSGSUB    ;REPORT ERROR
      054370  104456                                                                TRAP    C$ERHRD
      054372  001472                                                                .WORD   826
      054374  061010                                                                .WORD   T199CMP
      054376  013742                                                                .WORD   MSGSUB
 8386 054400                        210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      054400  104406                                                                TRAP    C$CLP1
 8387                               ;       Do a Write Subsystem READ STATUS
 8388                               ;       If Input Ready  NOT=1   Then Print Error
 8389                               ;       If Output Ready NOT=0   Then Print Error
 8390                               ;       If Data In Miss NOT=0   Then Print Error
 8391 054402  C04737  061330                JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
 8392 054406  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8393 054412  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8394 054416  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8395 054422                                FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
 8396 054436  103407                        BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
 8397 054440  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8398 054442                                NEXT.ERRNO
 8399 054442                        212$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      054442  104455                                                                TRAP    C$ERDF
      054444  001473                                                                .WORD   827
      054446  057674                                                                .WORD   T193SSR
      054450  012046                                                                .WORD   PKTSSR
 8400 054452  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8401 054456                        220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      054456  104406                                                                TRAP    C$CLP1
 8402 054460  004737  061570                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 8403 054464  012701  057432                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
 8404 054470  012702  061722                MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
 8405 054474  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 8406 054476  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 8407 054500  052711  000020                BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
 8408 054504  042711  000040                BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
 8409 054510  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
 8410 054514  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 8411 054516  012701  061702                MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8412 054522  012702  057412                MOV     #T19EXP,R2              ;EXPD ADDRESS
 8413 054526  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 8414 054532  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 8415 054536                                FORCERROR       232$,NOTSSR     ;@@D
 8416 054546  103404                        BCS     240$                    ;BR IF YES
 8417 054550                                NEXT.ERRNO
 8418 054550                        232$:   ERRHRD  ERRNO,T196CMP,MSGSTAT   ;REPORT ERROR
      054550  104456                                                                TRAP    C$ERHRD
      054552  001474                                                                .WORD   828
      054554  060550                                                                .WORD   T196CMP
      054556  012350                                                                .WORD   MSGSTAT
 8419 054560                        240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      054560  104406                                                                TRAP    C$CLP1
```

TEST 8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST

```
                                        FORCEXIT        255$            ;BBD
                                        MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
                                        CMP     R3,#TBLEND              ;END OF TSTBLK?
                                        BHIS    255$                    ;BR IF YES
                                        JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
                                255$:
                                        ENDSUB                          ;////////////// END SUBTEST //////////
                                                                                L10070:
                                                                                        TRAP    C$ESUB
                                        TST     FATFLG                  ;ANY FATAL ERRORS ?
                                        BEQ     260$                    ;BRANCH IF NOT
                                        JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
                                260$:

                                        .SBTTL  TEST  8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

                                ;++
                                ; TEST 8: SUBTEST 3:
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies the Write Strobe loopback path
                                ;       can strobe data from the FIFO to the Data lines.
                                ;       The signal IRESV3 drives IWSTR (write strobe) to write
                                ;       data from the FIFO to the tape data out latch.
                                ;
                                ; TEST STEPS:
                                ;
                                ;
                                ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
                                ;    BEGIN
                                ;       Write to TSSR register to soft initialize the controller
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
                                ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
                                ;       Do a WRITE NPR to set loopback and tape direction OUT
                                ;       Do a WRITE FORMAT to set IRESV3-->IWSTR = 1
                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
                                ;       Do a WRITE FORMAT to set IRESV3-->IWSTR = 0 to load write data latch
                                ;       Do a WRITE FORMAT to set IRESV3-->IWSTR = 1
                                ;       Do a WRITE NPR to set loopback and tape direction IN
                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
                                ;         to strobe loopback data into FIFO.
                                ;       Do a READ FIFO with tape direction IN to read data
                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
                                ;    END
                                ;--
                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                T8.3:
```

```
                054624  104402                                                          TRAP    C$BSUB
8474                                    ;       Write to TSSR register to soft initialize the controller
8475            054626                  5$:
8476            054626  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
8477            054632  103405          BCS     10$                             ;BR IF SOFT INIT OKAY
8478            054634  010001          MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
8479            054636                  ERRDF   ERRNO,SFIERR,SFIMSG             ;DEVICE FATAL DURING INIT
                054636  104455                                                          TRAP    C$ERDF
                054640  001474                                                          .WORD   828
                054642  003652                                                          .WORD   SFIERR
                054644  012034                                                          .WORD   SFIMSG
8480                                    ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
8481            054646  005037  002222  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
8482            054652  012704  061660          MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
8483            054656  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
8484            054662                          FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
8485            054676  103407          BCS     15$                             ;BR IF CARRY SET (GOOD RETURN)
8486            054700  010001          MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
8487            054702                  NEXT.ERRNO
8488            054702                  12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                054702  104455                                                          TRAP    C$ERDF
                054704  001475                                                          .WORD   829
                054706  057573                                                          .WORD   T19SSR
                054710  012046                                                          .WORD   PKTSSR
8489            054712  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
8490            054716                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                054716  104406                                                          TRAP    C$CLP1
8491                                    ;       If Extended Features Hardware Switch Clear then:
8492                                    ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
8493            054720  012701  061702          MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
8494            054724  032761  000200  000012  BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
8495            054732  001026          BNE     30$                             ;BR IF YES
8496            054734  004737  061532          JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
8497            054740  012704  062030          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8498            054744  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8499            054750  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8500            054754                          FORCERROR       22$                     ;@@FORCE ERROR IF FORCER=1
8501            054770  103407          BCS     30$                             ;BR IF CARRY SET (GOOD RETURN)
8502            054772  010001          MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
8503            054774                  NEXT.ERRNO
8504            054774                  22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                054774  104455                                                          TRAP    C$ERDF
                054776  001476                                                          .WORD   830
                055000  057630                                                          .WORD   T192SSR
                055002  012046                                                          .WORD   PKTSSR
8505            055004  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
8506            055010                  30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                055010  104406                                                          TRAP    C$CLP1
8507                                    ;       Do WRITE CHARACTERISTICS to select reserved unit 7
8508            055012  012704  061660          MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
8509            055016  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
8510            055022                          FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
8511            055036  103407          BCS     50$                             ;BR IF CARRY SET (GOOD RETURN)
8512            055040  010001          MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
8513            055042                  NEXT.ERRNO
8514            055042                  42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                055042  104455                                                          TRAP    C$ERDF
```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
        055044  001477                                                          .WORD   831
        055046  057573                                                          .WORD   T19SSR
        055050  012046                                                          .WORD   PKTSSR
  8515  055052  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
  8516  055056                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        055056  104406                                                          TRAP    C$CLP1
  8517
  8518                                  ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
  8519  055060  012703  002752                MOV     @TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
  8520  055064  012337  002312          100$: MOV     (R3)+,DATA              ;GET A TEST PATTERN
  8521  055070  042737  177400  002312      BIC     @#C<377>,DATA           ;DATA IS BYTE
  8522  055076  010337  002316              MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
  8523                                  ;       Do a WRITE NPR to set loopback and tape direction OUT
  8524  055102  012700  000100              MOV     @NP.OUT,R0              ;SET TAPE DIRECTION OUT
  8525  055106  052700  000040              BIS     @NP.LOOP,R0             ;SET LOOPBACK
  8526  055112  004737  061372              JSR     PC,T19SNPR              ;SETUP T19PK? FOR WRITE NPR
  8527  055116  012704  062030              MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
  8528  055122  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
  8529  055126  C04737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  8530  055132                              FORCERROR       102$                    ;@@FORCE ERROR IF FORCER=1
  8531  055146  103407                      BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
  8532  055150  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  8533  055152                              NEXT.ERRNO
  8534  055152                          102$: ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        055152  104455                                                          TRAP    C$ERDF
        055154  001500                                                          .WORD   832
        055156  057741                                                          .WORD   T194SSR
        055160  012046                                                          .WORD   PKTSSR
  8535  055162  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
  8536  055166                          105$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        055166  104406                                                          TRAP    C$CLP1
  8537                                  ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
  8538  055170  012700  000002              MOV     @WF.I3RES,R0            ;IRESV3==>IWSTR=1
  8539  055174  004737  061512              JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
  8540  055200  012704  062030              MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
  8541  055204  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
  8542  055210  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  8543  055214                              FORCERROR       112$                    ;@@FORCE ERROR IF FORCER=1
  8544  055230  103407                      BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
  8545  055232  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  8546  055234                              NEXT.ERRNO
  8547  055234                          112$: ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        055234  104455                                                          TRAP    C$ERDF
        055236  001501                                                          .WORD   833
        055240  060162                                                          .WORD   T198SSR
        055242  012046                                                          .WORD   PKTSSR
  8548  055244  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
  8549  055250                          120$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        055250  104406                                                          TRAP    C$CLP1
  8550                                  ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
  8551  055252  012700  000001              MOV     #1,R0                   ;WRITE 1 BYTE
  8552  055256  012701  002312              MOV     @DATA,R1                ;FIFO WRITE DATA ADDRESS
  8553  055262  004737  061436              JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
  8554  055266  012704  062030              MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
  8555  055272  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
  8556  055276  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  8557  055302                              FORCERROR       132$                    ;@@FORCE ERROR IF FORCER=1
```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
 8558 055316  103407                        BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
 8559 055320  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8560 055322                                NEXT.ERRNO
 8561 055322                        132$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055322  104455                                                                TRAP    C$ERDF
      055324  001502                                                                .WORD   834
      055326  060004                                                                .WORD   T195SSR
      055330  012046                                                                .WORD   PKTSSR
 8562 055332  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8563 055336                        140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055336  104406                                                                TRAP    C$CLP1
 8564                               ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 0
 8565 055340  005000                        CLR     R0                      ;SET IRESV3==>IWSTR=0
 8566 055342  004737  061512                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
 8567 055346  012704  062030                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8568 055352  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8569 055356  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8570 055362                                FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
 8571 055376  103407                        BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
 8572 055400  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8573 055402                                NEXT.ERRNO
 8574 055402                        152$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055402  104455                                                                TRAP    C$ERDF
      055404  001503                                                                .WORD   835
      055406  060162                                                                .WORD   T198SSR
      055410  012046                                                                .WORD   PKTSSR
 8575 055412  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8576 055416                        160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055416  104406                                                                TRAP    C$CLP1
 8577                               ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
 8578 055420  012700  000002                MOV     @WF.I3RES,R0            ;IRESV3==>IWSTR=1
 8579 055424  004737  061512                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
 8580 055430  012704  062030                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8581 055434  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8582 055440  004737  016336                JSR     PC,CHKTSSP              ;WAIT FOR SSR TO SET
 8583 055444                                FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
 8584 055460  103407                        BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
 8585 055462  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8586 055464                                NEXT.ERRNO
 8587 055464                        172$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055464  104455                                                                TRAP    C$ERDF
      055466  001504                                                                .WORD   836
      055470  060162                                                                .WORD   T198SSR
      055472  012046                                                                .WORD   PKTSSR
 8588 055474  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8589 055500                        180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055500  104406                                                                TRAP    C$CLP1
 8590
 8591                               ;       Do a WRITE NPR to set loopback and tape direction IN
 8592 055502  005000                        CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
 8593 055504  052700  000040                BIS     @NP.LOOP,R0             ;SET LOOPBACK
 8594 055510  004737  061372                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
 8595 055514  012704  062030                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8596 055520  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8597 055524  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8598 055530                                FORCERROR       182$                    ;@@DFORCE ERROR IF FORCER=1
 8599 055544  103407                        BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
 8600 055546  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8601 055550                                NEXT.ERRNO
 8602 055550                        182$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055550  104455                                                                TRAP    C$ERDF
      055552  001505                                                                .WORD   837
      055554  057741                                                                .WORD   T194SSR
      055556  012046                                                                .WORD   PKTSSR
 8603 055560  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8604 055564                        190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055564  104406                                                                TRAP    C$CLP1
 8605                               ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
 8606 055566  012700  000001                MOV     #1,R0                   ;WRITE 1 BYTE
 8607 055572  012701  002312                MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
 8608 055576  004737  061436                JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
 8609 055602  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8610 055606  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8611 055612  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8612 055616                                FORCERROR       202$                    ;@@FORCE ERROR IF FORCER=1
 8613 055632  103407                        BCS     210$                    ;BR IF CARRY SET (GOOD RETURN)
 8614 055634  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8615 055636                                NEXT.ERRNO
 8616 055636                        202$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055636  104455                                                                TRAP    C$ERDF
      055640  001506                                                                .WORD   838
      055642  060004                                                                .WORD   T195SSR
      055644  012046                                                                .WORD   PKTSSR
 8617 055646  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8618 055652                        210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055652  104406                                                                TRAP    C$CLP1
 8619                               ;       Do a READ FIFO with tape direction IN to read data
 8620 055654  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
 8621 055660  004737  061416                JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
 8622 055664  012704  062030                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8623 055670  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8624 055674  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8625 055700                                FORCERROR       222$                    ;@@FORCE ERROR IF FORCER=1
 8626 055714  103407                        BCS     230$                    ;BR IF CARRY SET (GOOD RETURN)
 8627 055716  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8628 055720                                NEXT.ERRNO
 8629 055720                        222$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055720  104455                                                                TRAP    C$ERDF
      055722  001507                                                                .WORD   839
      055724  060050                                                                .WORD   T196SSR
      055726  012046                                                                .WORD   PKTSSR
 8630 055730  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8631 055734                        230$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055734  104406                                                                TRAP    C$CLP1
 8632                               ;       If Data read from FIFO NOT= to Data sent Then Print Error
 8633 055736  004737  061570                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 8634 055742  012701  057432                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
 8635 055746  012702  061722                MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
 8636 055752  013711  002312                MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
 8637 055756  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
 8638 055764  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 8639 055766  012701  061702                MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8640 055772  012702  057412                MOV     #T19EXP,R2              ;EXPD ADDRESS
 8641 055776  012703  000022                MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
8642 056002  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
8643 056006                                    FORCERROR       242$,NOTSSR     ;SSD
8644 056016  103404                            BCS     250$                    ;BR IF YES
8645 056020                                    NEXT.ERRNO
8646 056020                            242$:   ERRHRD  ERRNO,T19WSTR,MSGSUB    ;REPORT ERROR
     056020  104456                                                                    TRAP    C$ERHRD
     056022  001510                                                                    .WORD   840
     056024  061073                                                                    .WORD   T19WSTR
     056026  013742                                                                    .WORD   MSGSUB
8647 056030                            250$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056030  104406                                                                    TRAP    C$CLP1
8648
8649
8650 056032                                    FORCEXIT        255$            ;SSD
8651 056042  013703  002316                    MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
8652 056046  020327  003062                    CMP     R3,#TBLEND              ;END OF TSTBLK?
8653 056052  103002                            BHIS    255$                    ;BR IF YES
8654 056054  000137  055064                    JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
8655 056060                            255$:
8656
8657 056060                                    ENDSUB                          ;////////////// END SUBTEST //////////
     056060                                                                        L10071:
     056060  104403                                                                    TRAP    C$ESUB
8658
8659 056062  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
8660 056066  001402                            BEQ     260$                    ;BRANCH IF NOT
8661 056070  004737  017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
8662 056074                            260$:
8663                                           .SBTTL  TEST  8: SUBTEST 4 LOOPBACK READ STROBE TEST
8664
8665                                   ;++
8666                                   ; TEST 8: SUBTEST 4:
8667                                   ;
8668                                   ; SUBTEST DESCRIPTION:
8669                                   ;
8670                                   ;       This subtest verifies the Read Strobe loopback path
8671                                   ;       can strobe the data from the Data lines to the FIFO.
8672                                   ;       The signal IRESV4 drives IRSTR (read strobe) to write
8673                                   ;       from the data lines to the FIFO.
8674                                   ;
8675                                   ; TEST STEPS:
8676                                   ;
8677                                   ;
8678                                   ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8679                                   ;    BEGIN
8680                                   ;       Write to TSSR register to soft initialize the controller
8681                                   ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
8682                                   ;       If Extended Features Hardware Switch Clear then:
8683                                   ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
8684                                   ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
8685                                   ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
8686                                   ;       Do a WRITE NPR to set loopback and tape direction OUT
8687                                   ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
8688                                   ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8689                                   ;       Do a READ FIFO with tape direction OUT to load tape out write latch
8690                                   ;       Do a WRITE NPR to set loopback and tape direction IN
8691                                   ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 to write loop data to FIFO
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
                                ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
                                ;         (to strobe loopback data into FIFO.)
                                ;       Do a READ FIFO with tape direction IN to read data
                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
                                ;   END
                                ;--
056074                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
056074                                                                  T8.4:
056074  104402                                                          TRAP    C$BSUB
                                ;       Write to TSSR register to soft initialize the controller
056076                          5$:
056076  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
056102  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
056104  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
056106                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
056106  104455                                                          TRAP    C$ERDF
056110  001510                                                          .WORD   840
056112  003652                                                          .WORD   SFIERR
056114  C12034                                                          .WORD   SFIMSG
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
056116  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
056122  012704  061660                  MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
056126  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
056132                                  FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
056146  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
056150  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
056152                                  NEXT.ERRNO
056152                          12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
056152  104455                                                          TRAP    C$ERDF
056154  001511                                                          .WORD   841
056156  057573                                                          .WORD   T19SSR
056160  012046                                                          .WORD   PKTSSR
056162  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
056166                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
056166  104406                                                          TRAP    C$CLP1
                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
056170  012701  061702          MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
056174  032761  000200  000012  BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
056202  001026                  BNE     30$                     ;BR IF YES
056204  004737  061532          JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
056210  012704  062030          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
056214  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
056220  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
056224                          FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
056240  103407                  BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
056242  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
056244                          NEXT.ERRNO
056244                          22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
056244  104455                                                          TRAP    C$ERDF
056246  001512                                                          .WORD   842
056250  057630                                                          .WORD   T192SSR
056252  012046                                                          .WORD   PKTSSR
056254  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
056260                          30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
056260  104406                                                          TRAP    C$CLP1
                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
8733 056262  012704  061660                MOV     @T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
8734 056266  004737  010662                JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
8735 056272                                FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
8736 056306  103407                        BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
8737 056310  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8738 056312                                NEXT.ERRNO
8739 056312                        42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     056312  104455                                                                TRAP    C$ERDF
     056314  001513                                                                .WORD   843
     056316  057573                                                                .WORD   T19SSR
     056320  012046                                                                .WORD   PKTSSR
8740 056322  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8741 056326                        50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056326  104406                                                                TRAP    C$CLP1
8742
8743                                ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8744 056330  012703  002752                MOV     @TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
8745 056334  012337  002312        100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
8746 056340  042737  177400  002312        BIC     @#C<377>,DATA           ;DATA IS BYTE
8747 056346  010337  002316                MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
8748                                ;       Do a WRITE NPR to set loopback and tape direction OUT
8749 056352  012700  000100                MOV     @#NP.OUT,R0             ;SET TAPE DIRECTION OUT
8750 056356  052700  000040                BIS     @#NP.LOOP,R0            ;SET LOOPBACK
8751 056362  004737  061372                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
8752 056366  012704  062030                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8753 056372  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8754 056376  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8755 056402                                FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
8756 056416  103407                        BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
8757 056420  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8758 056422                                NEXT.ERRNO
8759 056422                        102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056422  104455                                                                TRAP    C$ERDF
     056424  001514                                                                .WORD   844
     056426  057741                                                                .WORD   T194SSR
     056430  012046                                                                .WORD   PKTSSR
8760 056432  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8761 056436                        105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056436  104406                                                                TRAP    C$CLP1
8762                                ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
8763 056440  012700  000001                MOV     @WF.I4RES,R0            ;IRESV4==>IRSTR=1
8764 056444  004737  061512                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
8765 056450  012704  062030                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8766 056454  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8767 056460  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8768 056464                                FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
8769 056500  103407                        BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
8770 056502  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8771 056504                                NEXT.ERRNO
8772 056504                        112$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056504  104455                                                                TRAP    C$ERDF
     056506  001515                                                                .WORD   845
     056510  060162                                                                .WORD   T198SSR
     056512  012046                                                                .WORD   PKTSSR
8773 056514  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8774 056520                        120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056520  104406                                                                TRAP    C$CLP1
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
8775                                    ;       Do a WRITE FIFO with byte count equal to 1 and tape direction OUT
8776 056522  012700  000001                     MOV     #1,R0                   ;WRITE 1 BYTE
8777 056526  012701  002312                     MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
8778 056532  004737  061436                     JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
8779 056536  012704  062030                     MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8780 056542  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8781 056546  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8782 056552                                     FORCERROR       132$                    ;@@FORCE ERROR IF FORCER=1
8783 056566  103407                             BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
8784 056570  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8785 056572                                     NEXT.ERRNO
8786 056572                             132$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056572  104455                                                                     TRAP    C$ERDF
     056574  001516                                                                     .WORD   846
     056576  060004                                                                     .WORD   T195SSR
     056600  012046                                                                     .WORD   PKTSSR
8787 056602  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
8788 056606                             140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056606  104406                                                                     TRAP    C$CLP1
8789                                    ;       Do a READ FIFO with tape direction OUT to load tape out write latch
8790 056610  012700  000001                     MOV     #1,R0                   ;SET READ BYTE COUNT
8791 056614  004737  061416                     JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
8792 056620  012704  062030                     MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8793 056624  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8794 056630  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8795 056634                                     FORCERROR       152$                    ;@@FORCE ERROR IF FORCER=1
8796 056650  103407                             BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
8797 056652  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8798 056654                                     NEXT.ERRNO
8799 056654                             152$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056654  104455                                                                     TRAP    C$ERDF
     056656  001517                                                                     .WORD   847
     056660  060050                                                                     .WORD   T196SSR
     056662  012046                                                                     .WORD   PKTSSR
8800 056664  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
8801 056670                             160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056670  104406                                                                     TRAP    C$CLP1
8802                                    ;       Do a WRITE NPR to set loopback and tape direction IN
8803 056672  005000                             CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
8804 056674  052700  000040                     BIS     #NP.LOOP,R0             ;SET LOOPBACK
8805 056700  004737  061372                     JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
8806 056704  012704  062030                     MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8807 056710  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8808 056714  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8809 056720                                     FORCERROR       182$                    ;@@FORCE ERROR IF FORCER=1
8810 056734  103407                             BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
8811 056736  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8812 056740                                     NEXT.ERRNO
8813 056740                             182$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056740  104455                                                                     TRAP    C$ERDF
     056742  001520                                                                     .WORD   848
     056744  057741                                                                     .WORD   T194SSR
     056746  012046                                                                     .WORD   PKTSSR
8814 056750  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
8815 056754                             190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056754  104406                                                                     TRAP    C$CLP1
8816                                    ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 0
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
8817 056756  005000                    CLR     R0                      ;SET IRESV4==>IRSTR=0
8818 056760  004737  061512            JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
8819 056764  012704  062030            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8820 056770  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8821 056774  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8822 057000                            FORCERROR       202$                    ;@@DFORCE ERROR IF FORCER=1
8823 057014  103407                    BCS     210$                    ;BR IF CARRY SET (GOOD RETURN)
8824 057016  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8825 057020                            NEXT.ERRNO
8826 057020                    202$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057020  104455                                                                  TRAP    C$ERDF
     057022  001521                                                                  .WORD   849
     057024  060162                                                                  .WORD   T198SSR
     057026  012046                                                                  .WORD   PKTSSR
8827 057030  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
8828 057034                    210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057034  104406                                                                  TRAP    C$CLP1
8829                           ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
8830 057036  C12700  000001            MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR=1
8831 057042  004737  061512            JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
8832 057046  012704  062030            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8833 057052  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8834 057056  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8835 057062                            FORCERROR       222$                    ;@@DFORCE ERROR IF FORCER=1
8836 057076  103407                    BCS     230$                    ;BR IF CARRY SET (GOOD RETURN)
8837 057100  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8838 057102                            NEXT.ERRNO
8839 057102                    222$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057102  104455                                                                  TRAP    C$ERDF
     057104  001522                                                                  .WORD   850
     057106  060162                                                                  .WORD   T198SSR
     057110  012046                                                                  .WORD   PKTSSR
8840 057112  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
8841 057116                    230$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057116  104406                                                                  TRAP    C$CLP1
8842                           ;       Do a READ FIFO with tape direction IN to read data
8843 057120  012700  000001            MOV     #1,R0                   ;SET READ BYTE COUNT
8844 057124  004737  061416            JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
8845 057130  012704  062030            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8846 057134  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8847 057140  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8848 057144                            FORCERROR       282$                    ;@@DFORCE ERROR IF FORCER=1
8849 057160  103407                    BCS     290$                    ;BR IF CARRY SET (GOOD RETURN)
8850 057162  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8851 057164                            NEXT.ERRNO
8852 057164                    282$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057164  104455                                                                  TRAP    C$ERDF
     057166  001523                                                                  .WORD   851
     057170  060050                                                                  .WORD   T196SSR
     057172  012046                                                                  .WORD   PKTSSR
8853 057174  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
8854 057200                    290$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057200  104406                                                                  TRAP    C$CLP1
8855                           ;       If Data read from FIFO NOT= to Data sent Then Print Error
8856 057202  004737  061570            JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
8857 057206  012701  057432            MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
8858 057212  012702  061722            MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
 8859 057216  013711  002312                MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
 8860 057222  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
 8861 057230  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 8862 057232  012701  061702                MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8863 057236  012702  057412                MOV     #T19EXP,R2              ;EXPD ADDRESS
 8864 057242  012703  000022                MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
 8865 057246  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 8866 057252                                FORCERROR       302$,NOTSSR     ;SOD
 8867 057262  103404                        BCS     310$                    ;BR IF YES
 8868 057264                                NEXT.ERRNO
 8869 057264                        302$:   ERRHRD  ERRNO,T19RSTR,MSGSUB    ;REPORT ERROR
      057264  104456                                                                TRAP    C$ERHRD
      057266  001524                                                                .WORD   852
      057270  061200                                                                .WORD   T19RSTR
      057272  013742                                                                .WORD   MSGSUB
 8870 057274                        310$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      057274  104406                                                                TRAP    C$CLP1
 8871
 8872
 8873 057276                                FORCEXIT        355$            ;SOD
 8874 057306  013703  002316                MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
 8875 057312  020327  003062                CMP     R3,#TBLEND              ;END OF TSTBLK?
 8876 057316  103002                        BHIS    355$                    ;BR IF YES
 8877 057320  000137  056334                JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
 8878 057324                        355$:
 8879
 8880 057324                                ENDSUB                          ;////////////// END SUBTEST //////////
      057324                                                                    L10072:
      057324  104403                                                                TRAP    C$ESUB
 8881
 8882 057326  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
 8883 057332  001402                        BEQ     360$                    ;BRANCH IF NOT
 8884 057334  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 8885 057340                        360$:
 8886
 8887 057340                        EXIT    TST                             ;////////// EXIT TEST //////////
      057340  104432                                                                TRAP    C$EXIT
      057342  002602                                                                .WORD   L10066-.
 8888
 8889
 8890                               ;+
 8891                               ;LOCAL STORAGE FOR THIS TEST
 8892                               ;-
 8893
 8894 057344  000000                T19PREV:        .WORD   0               ;DRIVE SIGNAL 1-0 TRANSITION FLAG
 8895                               ;+
 8896                               ; LOOPBACK DRIVE SIGNAL TABLE
 8897                               ; THIS TABLE IS USED BY T19CNVT TO SETUP
 8898                               ; A DRIVE PATTERN FROM THE TEST DATA INPUT PATTERN.
 8899                               ;
 8900                               ; WRITE CONTROL SIGNALS ARE OF FORM WC.XXX
 8901                               ; WRITE FORMAT SIGNALS ARE OF FORM WF.XXXX
 8902                               ;-
 8903 057346                        T19BFCTL:                               ;WRITE CONTROL DRIVE SIGNALS
 8904 057346  000001                        WC.IGO                          ;IGO==>IFPT     DATA<0>
 8905 057350  000002                        WC.IFEN                         ;IFEN==>IFBY    DATA<1>
 8906 057352  000004                        WC.IRWU                         ;IRWU==>IRWD    DATA<2>
```

```
 8907 057354  000010                              WC.IREW                 ;IREW==>IDBY    DATA<3>
 8908 057356  002000                              WF.IERASE*256.          ;IFAD==>ILDP    DATA<4>
 8909 057360  000040                              WC.I1TAD                ;ITAD1==>IONL   DATA<5>
 8910 057362  000100                              WC.IOTAD                ;ITADO==>IRDY   DATA<6>
 8911 057364  000200                              WC.IFAD                 ;IERASE==>ISPEED DATA<7>
 8912 057366  004000                              WF.IEDIT*256.           ;IEDIT==>IHER   DATA<8>
 8913 057370  010000                              WF.IWFM*256.            ;IWFM==>IFMK    DATA<9>
 8914 057372  020000                              WF.IREV*256.            ;IREV==>ICER    DATA<10>
 8915 057374  040000                              WF.IWRT*256.            ;IWRT==>IIDENT  DATA<11>
 8916 057376  100000                              WF.IHISP*256.           ;IHISP==>IEOT   DATA<12>
 8917 057400  000000                              .WORD   0               ;IRESV2 (UNUSED)DATA<13>
 8918 057402  000000                              .WORD   0               ;IRESV1 (UNUSED)DATA<14>
 8919 057404  000000                              .WORD   0               ;PARERR (UNTESTED)DATA<15>
 8920
 8921 057406                          T19MSK:                             ;MASK OF UNTESTED BITS IN READ STATUS BYTES
 8922                                                                     ;UNTESTED BITS ARE SET TO 1
 8923 057406      377                         .BYTE   ^C<000>             ;BYTE 0 MASK
 8924 057407      037                         .BYTE   ^C<340>             ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
 8925 057410      360                         .BYTE   ^C<017>             ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
 8926 057411      000                         .BYTE   0                   ;MAKE IT EVEN
 8927
 8928 057412                          T19EXP:                             ;BEGIN EXPECTED DATA BUFFER
 8929 057412  000000                         .WORD   0                   ;MESSAGE TYPE
 8930 057414  000000                         .WORD   0                   ;DATA FIELD LENGTH
 8931 057416  000000                         .WORD   0                   ;RBPCR
 8932 057420  000000                         .WORD   0                   ;XST0
 8933 057422  000000                         .WORD   0                   ;XST1
 8934 057424  000000                         .WORD   0                   ;XST2
 8935 057426  000000                         .WORD   0                   ;XST3
 8936 057430  000000                         .WORD   0                   ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
 8937 057432                          T19EXSTA: .BLKB 64.                ;EXPECTED READ STATUS AND WRITE FIFO DATA
 8938 057532                          T19EXEND:                                  ;END EXPECTED DATA BUFFER
 8939                                 ;+
 8940                                 ;LOCAL TEXT MESSAGES FOR TEST
 8941                                 ;-
 8942
 8943 057532  124  162  141  TST19ID:          .ASCIZ  'Transport Bus Interface Loopback'
 8944 057573  127  122  111  T19SSR:  .ASCIZ  'WRITE CHARACTERISTICS Failed'
 8945 057630  127  122  111  T192SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
 8946 057674  127  122  111  T193SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
 8947 057741  127  122  111  T194SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
 8948 060004  127  122  111  T195SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
 8949 060050  127  122  111  T196SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
 8950 060113  127  122  111  T197SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Control) Failed'
 8951 060162  127  122  111  T198SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Format) Failed'
 8952 060230  106  111  106  T191CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize'
 8953 060312  122  145  141  T192CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
 8954 060406  124  141  160  T193CMP:.ASCIZ  'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
 8955 060474  122  145  141  T195CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
 8956 060550  106  111  106  T196CMP:.ASCIZ  'FIFO Status (in WORD #9) Incorrect after READ FIFO'
 8957 060633  124  141  160  T197CMP:.ASCIZ  'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
 8958 060721  103  157  156  T198CMP:.ASCIZ  'Control Signal Loopback Data Error, Data is in WORD #8'
 8959 061010  122  145  141  T199CMP:.ASCIZ  'Read/Write Loopback Data Error, Data is in WORD #8'
 8960 061073  114  157  157  T19WSTR:.ASCIZ  'Loopback Data Error when strobed by Write strobe, Data is in WORD #8'
 8961 061200  114  157  157  T19RSTR:.ASCIZ  'Loopback Data Error when strobed by Read Strobe, Data is in WORD #8'
 8962
 8963                                         .EVEN
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
8964
8965                                    ;+
8966                                    ; CLEAR MESSAGE BUFFER
8967                                    ;-
8968 061304                             T19CLRBUF:                              ;
8969 061304                                     SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
8970 061310  012701  061702                     MOV     #T19BFR,R1              ;GET MESSAGE BUFFER ADDRESS
8971 061314  012702  000120                     MOV     #T19BEND-T19BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES
8972 061320  105021                     10$:    CLRB    (R1)+                   ;CLEAR A BYTE
8973 061322  005302                             DEC     R2                      ;DONE?
8974 061324  003375                             BGT     10$                     ;BR IF NO
8975 061326  000207                             RTS     PC                      ;RETURN
8976
8977                                    ;+
8978                                    ; SETUP T19PK2 PACKET FOR READ STATUS
8979                                    ;-
8980 061330                             T19SRD:
8981 061330  004737  061304                     JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
8982 061334  012700  062040                     MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
8983 061340  112720  000005                     MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
8984 061344  105010                             CLRB    (R0)                    ;CLEAR BSEL1
8985 061346  000207                             RTS     PC                      ;RETURN
8986
8987                                    ;+
8988                                    ; SETUP T19PK2 PACKET FOR WRITE MISC Reset Tape Status F-FLOPS
8989                                    ;-
8990 061350                             T19RSFIF:
8991 061350  004737  061304                     JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
8992 061354  012700  062040                     MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
8993 061360  112720  000010                     MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
8994 061364  112710  000030                     MOVB    #MS.RSFIF!MS.RSTAP,(R0) ;STORE BSEL1 CLEAR FIFO CODES
8995 061370  000207                             RTS     PC                      ;RETURN
8996
8997                                    ;+
8998                                    ; SETUP T19PK2 PACKET FOR WRITE NPR
8999                                    ;
9000                                    ; INPUT:
9001                                    ;       R0 CONTAINS BSEL1 NPR DATA
9002                                    ;
9003                                    ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
9004                                    ;-
9005 061372                             T19SNPR:
9006 061372  004737  061304                     JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
9007 061376  012701  062040                     MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
9008 061402  112721  000011                     MOVB    #PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
9009 061406  052700  000020                     BIS     #NP.WRP,R0              ;DON'T WRITE WRONG PARITY
9010 061412  110011                             MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
9011 061414  000207                             RTS     PC                      ;RETURN
9012
9013                                    ;+
9014                                    ; SETUP T19PK2 PACKET FOR READ FIFO
9015                                    ;
9016                                    ; INPUT:
9017                                    ;       R0 CONTAINS SEL2 BYTE COUNT
9018                                    ;-
9019 061416                             T19RFIF:
9020 061416  004737  061304                     JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
9021 061422  012701  062040                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
9022 061426  112721  000003                MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSELO
9023 061432  110021                        MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
9024 061434  000207                        RTS     PC                      ;RETURN
9025                                ;+
9026                                ; SETUP T19PK2 PACKET FOR WRITE FIFO
9027                                ;
9028                                ; INPUT:
9029                                ;       R0 CONTAINS BYTE COUNT
9030                                ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
9031                                ;-
9032 061436                        T19WFIF:
9033 061436                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
9034 061442  004737  061304                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
9035 061446  012702  062040                MOV     #T19DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
9036 061452  112722  000004                MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSELO
9037 061456  110022                        MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
9038 061460  005022                        CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
9039 061462  112122                10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
9040 061464  005300                        DEC     R0                      ;DONE ALL BYTES?
9041 061466  003375                        BGT     10$                     ;BR IF NO
9042 061470  000207                        RTS     PC                      ;RETURN
9043                                ;+
9044                                ; SETUP T19PK2 FOR WRITE CONTROL
9045                                ;
9046                                ; INPUT:
9047                                ;       R0 CONTAINS DRIVING DATA PATTERN
9048                                ;-
9049 061472                        T19WCTL:
9050 061472  004737  061304                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
9051 061476  012701  062040                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
9052 061502  112721  000006                MOVB    #PW.WCTL,(R1)+          ;STORE WRITE CONTROL IN BSELO
9053 061506  110021                        MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
9054 061510  000207                        RTS     PC                      ;RETURN
9055                                ;+
9056                                ; SETUP T19PK2 FOR WRITE FORMAT TRANSPORT REGISTER
9057                                ;
9058                                ; INPUT:
9059                                ;       R0 CONTAINS DRIVING DATA PATTERN
9060                                ;-
9061 061512                        T19WFMT:
9062 061512  004737  061304                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
9063 061516  012701  062040                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
9064 061522  112721  000007                MOVB    #PW.WFMT,(R1)+          ;STORE WRITE FORMAT IN BSELO
9065 061526  110021                        MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
9066 061530  000207                        RTS     PC                      ;RETURN
9067                                ;+
9068                                ; SETUP T19PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
9069                                ;-
9070 061532                        T19SEXT:
9071 061532  012700  062040                MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
9072 061536  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSELO
9073 061542  112710  000200                MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
9074 061546  000207                        RTS     PC                      ;RETURN
9075                                ;+
9076                                ; CLEAR EXPECTED DATA MESSAGE BUFFER
9077                                ;-
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
9078 061550                            T19CLEXP:
9079 061550  012701  057412                MOV     #T19EXP,R1              ;GET EXPD ADDRESS
9080 061554  012700  000120                MOV     #T19EXEND-T19EXP,R0     ;GET EXPD SIZE
9081 061560  105021                10$:    CLRB    (R1)+                   ;CLEAR A BYTE
9082 061562  005300                        DEC     R0                      ;DONE?
9083 061564  003375                        BGT     10$                     ;BR IF NO
9084 061566  000207                        RTS     PC                      ;RETURN
9085
9086                                   ;+
9087                                   ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
9088                                   ;-
9089 061570                            T19SETEXP:
9090 061570  012702  057412                MOV     #T19EXP,R2              ;GET EXPD
9091 061574  012703  061702                MOV     #T198FR,R3              ;GET READ STATUS RECV BUFFER
9092 061600  012700  000010                MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
9093 061604  012322                5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
9094 061606  005300                        DEC     R0                      ;DONE WORDS 0-7 WORDS?
9095 061610  003375                        BGT     5$                      ;BR IF NO
9096 061612  000207                        RTS     PC                      ;RETURN
9097
9098                                   ;+
9099                                   ; CONVERT A TEST PATTERN DATA WORD TO LOOPBACK DRIVE SIGNALS
9100                                   ;
9101                                   ; INPUTS:
9102                                   ;
9103                                   ;       R0      TEST PATTERN
9104                                   ;
9105                                   ; IMPLICIT INPUTS:
9106                                   ;
9107                                   ;       T198FCTL  - CONTAINS WRITE CONTROL / WRITE FORMAT CONVERSION BITS
9108                                   ;
9109                                   ; OUTPUTS:
9110                                   ;
9111                                   ;       R0      - LOW BYTE CONTAINS WRITE CONTROL DATA
9112                                   ;               - HIGH BYTE CONTAINS WRITE FORMAT DATA
9113                                   ;-
9114 061614                            T19CNVT:
9115 061614                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
9116 061620  012701  057346                MOV     #T198FCTL,R1            ;CONVERSION TABLE ADDRESS
9117 061624  005002                        CLR     R2                      ;INIT RESULT OF CONVERSION
9118 061626  012703  000020                MOV     #16.,R3                 ;BIT COUNT
9119 061632  006000                10$:    ROR     R0                      ;IS THIS BIT EQUAL TO 1?
9120 061634  103001                        BCC     20$                     ;BR IF NO
9121 061636  051102                        BIS     (R1),R2                 ;SET CONVERTED BIT
9122 061640  005721                20$:    TST     (R1)+                   ;POINT TO NEXT BIT IN CONVERSION TABLE
9123 061642  005303                        DEC     R3                      ;DONE?
9124 061644  003372                        BGT     10$                     ;BR IF NO
9125 061646  010200                        MOV     R2,R0                   ;COPY RESULT
9126 061650  000207                        RTS     PC                      ;RETURN
9127
9128
9129
9131         061660                        .=<.+10>&177770
9133                                   ;
9134                                   ;WRITE CHARACTERISTICS COMMAND PACKET
9135                                   ;
9136 061660                            T19PACKET:                              ;COMMAND PACKET FOR TEST
```

```
TEST  8: SUBTEST 4 LOOPBACK READ STROBE TEST

 9137 061660  100004                        .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
 9138 061662  061670                        .WORD   T19DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
 9139 061664  000000                        .WORD   0
 9140 061666  000012                        .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 9141
 9142 061670                        T19DATA:                                ;CHARACTERISTICS DATA BLOCK
 9143 061670  061702                        .WORD   T19BFR                  ;ADDRESS OF MESSAGE BUFFER
 9144 061672  000000                        .WORD   0
 9145 061674  000024                        .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
 9146 061676  000000                        .WORD   0                       ;ESS,ENB,EAI,ERI
 9147 061700  000007                        .WORD   7                       ;EXTENDED FEATURES UNIT NO.
 9148
 9149
 9150                               ;MESSAGE BUFFER FOR ALL TEST 8 COMMANDS
 9151
 9152 061702                        T19BFR:                                 ;BEGIN MESSAGE BUFFER
 9153 061702  000000                        .WORD   0                       ;MESSAGE TYPE
 9154 061704  000000                        .WORD   0                       ;DATA FIELD LENGTH
 9155 061706  C00000                        .WORD   0                       ;RBPCR
 9156 061710  000000                        .WORD   0                       ;XST0
 9157 061712  000000                        .WORD   0                       ;XST1
 9158 061714  000000                        .WORD   0                       ;XST2
 9159 061716  000000                        .WORD   0                       ;XST3
 9160 061720  000000                        .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
 9161 061722                        T19BFSTA: .BLKB 64.                     ;READ STATUS AND WRITE FIFO BUFFER
 9162 062022                        T19BEND:                                ;END OF MESSAGE BUFFER
 9163                               ;
 9164                               ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
 9165                               ;
 9167         062030                        .=<.+10>&177770
 9169 062030                        T19PK2:
 9170 062030  100006                        .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
 9171 062032  062040                        .WORD   T19DT2                  ;LOW ADDRESS OF DATA BLOCK
 9172 062034  000000                        .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
 9173 062036  000012                        .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 9174
 9175 062040                        T19DT2:                                 ;DATA BLOCK
 9176 062040     000                        .BYTE   0                       ;BSEL0
 9177 062041     000                        .BYTE   0                       ;BSEL1
 9178 062042  000000                        .WORD   0                       ;SEL2
 9179 062044                                .BLKB   64.                     ;WRITE FIFO DATA OUTPUT BUFFER
 9180
 9181
 9182 062144                                ENDTST
      062144                                                          L10066:
      062144  104401                                                           TRAP    C$ETST
 9183                                       .SBTTL  TEST  9: READ/WRITE DATA PARITY TEST
 9184                               ;++
 9185                               ; TEST DESCRIPTION:
 9186                               ;
 9187                               ;       This test verifies that the Write Data Parity generator
 9188                               ;       and the Read Data Parity checker operate properly. The
 9189                               ;       Transport Bus signal loopback mode is enabled and a
 9190                               ;       Set Wrong parity function is executed. Then various
 9191                               ;       Write Subsystem Memory functions are performed to
 9192                               ;       write data to and from the FIFO in loopback mode.
 9193                               ;       The program then checks to insure a Read Data parity
```

```
                                ;       error occurred.
                                ;       A Reset FIFO is done and the Read Data parity
                                ;       error bit is again tested to insure it cleared.
                                ;       Finally a Clear wrong parity function is done
                                ;       and it is verified the data word can pass in loopback
                                ;       mode without setting Read Data parity error.
                                ;
                                ; TEST STEPS:
                                ;
                                ;  REPEAT FOR LOOPCNT
                                ;    BEGIN
                                ;       Write to TSSR register to soft initialize the controller
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
                                ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
                                ;    BEGIN
                                ;       (* Verify Write Wrong Parity Sets Parity Error *)
                                ;       Do a WRITE NPR to set loopback and tape direction OUT
                                ;              and SET Write Wrong Parity.
                                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
                                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
                                ;         (this is when wrong parity (IWP) is set)
                                ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)
                                ;            (Read Strobe sets PAR IN H [Parity Error])
                                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Read Data parity error NOT=1  Then Print Error
                                ;       Do a Write Misc to RESET FIFO
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Read Data parity error NOT=0  Then Print Error
                                ;
                                ;       (* Verify Data can be transferred without a Parity Error *)
                                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
                                ;       Do a WRITE NPR to set loopback and tape direction OUT
                                ;            and CLEAR Write Wrong Parity.
                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
                                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
                                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
                                ;            (Read Strobe should NOT set PAR IN H [Parity Error] here)
                                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Read Data parity error NOT=0  Then Print Error
                                ;
                                ;    END
                                ;--
                                
062146                                  BGNTST
062146                                                                          T9::
062146  012700  064532                  MOV     #TST20ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
062152  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
062156  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
062164                          T20LOOP:
```

TEST 9: READ/WRITE DATA PARITY TEST

```
9254 062164                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////////
     062164                                                                                      T9.1:
     062164  104402                                                                            TRAP    C$BSUB
9255                                   ;       Write to TSSR register to soft initialize the controller
9256 062166                            5$:
9257 062166  004737  015774                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
9258 062172  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
9259 062174  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9260 062176                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     062176  104455                                                                            TRAP    C$ERDF
     062200  001604                                                                            .WORD   900
     062202  003652                                                                            .WORD   SFIERR
     062204  012034                                                                            .WORD   SFIMSG
9261                                   ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
9262 062206  005037  002222            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
9263 062212  012704  065730                    MOV     #T20PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
9264 062216  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
9265 062222                                    FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
9266 062236  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
9267 062240  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9268 062242                                    NEXT.ERRNO
9269 062242                            12$:    ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     062242  104455                                                                            TRAP    C$ERDF
     062244  001605                                                                            .WORD   901
     062246  064561                                                                            .WORD   T20SSR
     062250  012046                                                                            .WORD   PKTSSR
9270 062252  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
9271 062256                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     062256  104406                                                                            TRAP    C$CLP1
9272                                   ;       If Extended Features Hardware Switch Clear then:
9273                                   ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
9274 062260  012701  065752                    MOV     #T20BFR,R1              ;MESSAGE BUFFER ADDRESS
9275 062264  032761  000200  000012            BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
9276 062272  001026                            BNE     30$                     ;BR IF YES
9277 062274  004737  065646                    JSR     PC,T20SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
9278 062300  012704  066100                    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9279 062304  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9280 062310  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9281 062314                                    FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
9282 062330  103407                            BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
9283 062332  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9284 062334                                    NEXT.ERRNO
9285 062334                            22$:    ERRDF   ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     062334  104455                                                                            TRAP    C$ERDF
     062336  001606                                                                            .WORD   902
     062340  064616                                                                            .WORD   T202SSR
     062342  012046                                                                            .WORD   PKTSSR
9286 062344  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
9287 062350                            30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     062350  104406                                                                            TRAP    C$CLP1
9288                                   ;       Do WRITE CHARACTERISTICS to select reserved unit 7
9289 062352  012704  065730                    MOV     #T20PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
9290 062356  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
9291 062362                                    FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
9292 062376  103407                            BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
9293 062400  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9294 062402                                    NEXT.ERRNO
```

TEST 9: READ/WRITE DATA PARITY TEST

```
 9295 062402                            42$:    ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      062402  104455                                                                    TRAP    C$ERDF
      062404  001607                                                                    .WORD   903
      062406  064561                                                                    .WORD   T20SSR
      062410  012046                                                                    .WORD   PKTSSR
 9296 062412  005237  002222                       INC     FATFLG                  ;SET FATAL ERROR FLAG
 9297 062416                            50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      062416  104406                                                                    TRAP    C$CLP1
 9298
 9299
 9300                                   ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
 9301 062420  012703  002752                       MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
 9302 062424  012337  002312            100$:    MOV     (R3)+,DATA              ;GET A TEST PATTERN
 9303 062430  042737  177400  002312              BIC     #^C<377>,DATA           ;DATA IS BYTE
 9304 062436  010337  002316                       MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
 9305                                   ;       Do a WRITE NPR to set loopback and tape direction OUT and
 9306                                   ;                 and SET Write Wrong Parity.
 9307 062442  012700  000100                       MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
 9308 062446  C52700  000040                       BIS     #NP.LOOP,R0             ;SET LOOPBACK
 9309 062452  042700  000020                       BIC     #NP.WRP,R0              ;SET WRITE WRONG PARITY (INVERTED)
 9310 062456  004737  065516                       JSR     PC,T20WNPR              ;SETUP T20PK2 FOR WRITE NPR
 9311 062462  012704  066100                       MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9312 062466  010465  000000                       MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9313 062472  004737  016336                       JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9314 062476                                       FORCERROR       102$                    ;@@FORCE ERROR IF FORCER=1
 9315 062512  103407                               BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
 9316 062514  010001                               MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9317 062516                                       NEXT.ERRNO
 9318 062516                            102$:    ERRDF   ERRNO,T204SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      062516  104455                                                                    TRAP    C$ERDF
      062520  001610                                                                    .WORD   904
      062522  064727                                                                    .WORD   T204SSR
      062524  012046                                                                    .WORD   PKTSSR
 9319 062526  005237  002222                       INC     FATFLG                  ;SET FATAL ERROR FLAG
 9320 062532                            105$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      062532  104406                                                                    TRAP    C$CLP1
 9321                                   ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
 9322 062534  012700  000001                       MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
 9323 062540  004737  065612                       JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
 9324 062544  012704  066100                       MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9325 062550  010465  000000                       MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9326 062554  004737  016336                       JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9327 062560                                       FORCERROR       112$                    ;@@FORCE ERROR IF FORCER=1
 9328 062574  103407                               BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
 9329 062576  010001                               MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9330 062600                                       NEXT.ERRNO
 9331 062600                            112$:    ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      062600  104455                                                                    TRAP    C$ERDF
      062602  001611                                                                    .WORD   905
      062604  065101                                                                    .WORD   T208SSR
      062606  012046                                                                    .WORD   PKTSSR
 9332 062610  005237  002222                       INC     FATFLG                  ;SET FATAL ERROR FLAG
 9333 062614                            120$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      062614  104406                                                                    TRAP    C$CLP1
 9334                                   ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
 9335 062616  012700  000001                       MOV     #1,R0                   ;WRITE 1 BYTE
 9336 062622  012701  002312                       MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
```

TEST 9: READ/WRITE DATA PARITY TEST

```
 9337 062626  004737  065556                  JSR     PC,T20WFIF              ;SETUP T20PK2 FOR WRITE FIFO
 9338 062632  012704  066100                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9339 062636  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9340 062642  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9341 062646                                  FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
 9342 062662  103407                          BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
 9343 062664  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9344 062666                                  NEXT.ERRNO
 9345 062666                          152$:   ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      062666  104455                                                                      TRAP    C$ERDF
      062670  001612                                                                      .WORD   906
      062672  064772                                                                      .WORD   T20SSR
      062674  012046                                                                      .WORD   PKTSSR
 9346 062676  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 9347 062702                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      062702  104406                                                                      TRAP    C$CLP1
 9348                                 ;       Do a READ FIFO with tape direction OUT to load tape out write latch
 9349                                 ;         (this is when wrong parity (IWP) is set)
 9350 062704  C12700  000001                  MOV     #1,R0                   ;SET READ BYTE COUNT
 9351 062710  004737  065536                  JSR     PC,T20RFIF              ;SETUP T20PK2 FOR READ FIFO
 9352 062714  012704  066100                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9353 062720  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9354 062724  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9355 062730                                  FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
 9356 062744  103407                          BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
 9357 062746  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9358 062750                                  NEXT.ERRNO
 9359 062750                          172$:   ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      062750  104455                                                                      TRAP    C$ERDF
      062752  001613                                                                      .WORD   907
      062754  065036                                                                      .WORD   T20SSR
      062756  012046                                                                      .WORD   PKTSSR
 9360 062760  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 9361 062764                          180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      062764  104406                                                                      TRAP    C$CLP1
 9362                                 ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
 9363                                 ;               (Read Strobe sets PAR IN H [Parity Error])
 9364 062766  005000                          CLR     R0                      ;IRESV4==>IRSTR = 0
 9365 062770  004737  065612                  JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
 9366 062774  012704  066100                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9367 063000  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9368 063004  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9369 063010                                  FORCERROR       192$                    ;@@DFORCE ERROR IF FORCER=1
 9370 063024  103407                          BCS     200$                    ;BR IF CARRY SET (GOOD RETURN)
 9371 063026  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9372 063030                                  NEXT.ERRNO
 9373 063030                          192$:   ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      063030  104455                                                                      TRAP    C$ERDF
      063032  001614                                                                      .WORD   908
      063034  065101                                                                      .WORD   T20SSR
      063036  012046                                                                      .WORD   PKTSSR
 9374 063040  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 9375 063044                          200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      063044  104406                                                                      TRAP    C$CLP1
 9376                                 ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
 9377 063046  012700  000001                  MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
 9378 063052  004737  065612                  JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
```

TEST 9: READ/WRITE DATA PARITY TEST

```
9379 063056  012704  066100                    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9380 063062  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9381 063066  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9382 063072                                    FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
9383 063106  103407                            BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
9384 063110  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9385 063112                                    NEXT.ERRNO
9386 063112                            212$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063112  104455                                                                  TRAP    C$ERDF
     063114  001615                                                                  .WORD   909
     063116  065101                                                                  .WORD   T208SSR
     063120  012046                                                                  .WORD   PKTSSR
9387 063122  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
9388 063126                            220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063126  104406                                                                  TRAP    C$CLP1
9389                                   ;       Do a Write Subsystem READ STATUS
9390 063130  004737  065476                    JSR     PC,T20SRD               ;SETUP PACKET FOR READ STATUS
9391 063134  012704  066100                    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9392 063140  C10465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9393 063144  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9394 063150                                    FORCERROR       232$                    ;@@DFORCE ERROR IF FORCER=1
9395 063164  103407                            BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
9396 063166  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9397 063170                                    NEXT.ERRNO
9398 063170                            232$:   ERRDF   ERRNO,T203SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063170  104455                                                                  TRAP    C$ERDF
     063172  001616                                                                  .WORD   910
     063174  064662                                                                  .WORD   T203SSR
     063176  012046                                                                  .WORD   PKTSSR
9399 063200  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
9400 063204                            240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063204  104406                                                                  TRAP    C$CLP1
9401                                   ;       If Read Data parity error NOT=1  Then Print Error
9402 063206  004737  065704                    JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
9403 063212  012701  064432                    MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
9404 063216  012702  065772                    MOV     #T20BFSTA,R2            ;GET RECV READ STATUS
9405 063222  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
9406 063224  016261  000002  000002            MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
9407 063232  052711  100000                    BIS     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =1
9408 063236  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
9409 063240  012701  065752                    MOV     #T20BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
9410 063244  012702  064412                    MOV     #T20EXP,R2              ;EXPD ADDRESS
9411 063250  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
9412 063254  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
9413 063260                                    FORCERROR       252$,NOTSSR             ;@@D
9414 063270  103404                            BCS     260$                    ;BR IF YES
9415 063272                                    NEXT.ERRNO
9416 063272                            252$:   ERRHRD  ERRNO,T20SWP,MSGSTAT    ;REPORT ERROR
     063272  104456                                                                  TRAP    C$ERHRD
     063274  001617                                                                  .WORD   911
     063276  065147                                                                  .WORD   T20SWP
     063300  012350                                                                  .WORD   MSGSTAT
9417 063302                            260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063302  104406                                                                  TRAP    C$CLP1
9418                                   ;       Do a Write Misc to RESET FIFO
9419 063304  012700  000020                    MOV     #MS.RSFIF,R0            ;SET RESET FIFO COMMAND
9420 063310  004737  065632                    JSR     PC,T20WMISC             ;SETUP T20PK2 FOR WRITE MISC
```

TEST 9: READ/WRITE DATA PARITY TEST

```
9421 063314  012704  066100                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9422 063320  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9423 063324  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9424 063330                                FORCERROR       282$                    ;@@DFORCE ERROR IF FORCER=1
9425 063344  103407                        BCS     290$                    ;BR IF CARRY SET (GOOD RETURN)
9426 063346  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9427 063350                                NEXT.ERRNO
9428 063350                        282$:   ERRDF   ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063350  104455                                                                TRAP    C$ERDF
     063352  001620                                                                .WORD   912
     063354  064616                                                                .WORD   T202SSR
     063356  012046                                                                .WORD   PKTSSR
9429 063360  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
9430 063364                        290$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063364  104406                                                                TRAP    C$CLP1
9431                                ;       Do a Write Subsystem READ STATUS
9432                                ;       If Read Data parity error NOT=0  Then Print Error
9433 063366  004737  065704                JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
9434 063372  012701  064432                MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
9435 063376  012702  065772                MOV     #T20BFSTA,R2            ;GET RECV READ STATUS
9436 063402  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
9437 063404  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
9438 063412  042711  100000                BIC     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =0
9439 063416  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
9440 063420  012701  065752                MOV     #T20BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
9441 063424  012702  064412                MOV     #T20EXP,R2              ;EXPD ADDRESS
9442 063430  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
9443 063434  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
9444 063440                                FORCERROR       302$,NOTSSR             ;@@D
9445 063450  103404                        BCS     320$                    ;BR IF YES
9446 063452                                NEXT.ERRNO
9447 063452                        302$:   ERRHRD  ERRNO,T20RSF,MSGSTAT    ;REPORT ERROR
     063452  104456                                                                TRAP    C$ERHRD
     063454  001621                                                                .WORD   913
     063456  065256                                                                .WORD   T20RSF
     063460  012350                                                                .WORD   MSGSTAT
9448 063462                        320$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063462  104406                                                                TRAP    C$CLP1
9449                                ;       (* Verify Data can be transferred without a Parity Error *)
9450                                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
9451 063464  012700  000001                MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
9452 063470  004737  065612                JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
9453 063474  012704  066100                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9454 063500  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9455 063504  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9456 063510                                FORCERROR       332$                    ;@@DFORCE ERROR IF FORCER=1
9457 063524  103407                        BCS     340$                    ;BR IF CARRY SET (GOOD RETURN)
9458 063526  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9459 063530                                NEXT.ERRNO
9460 063530                        332$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063530  104455                                                                TRAP    C$ERDF
     063532  001622                                                                .WORD   914
     063534  065101                                                                .WORD   T208SSR
     063536  012046                                                                .WORD   PKTSSR
9461 063540  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
9462 063544                        340$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063544  104406                                                                TRAP    C$CLP1
```

TEST 9: READ/WRITE DATA PARITY TEST

```
9463                                    ;       Do a WRITE NPR to set loopback and tape direction OUT and
9464                                    ;               and CLEAR Write Wrong Parity.
9465 063546  012700  000100              MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
9466 063552  052700  000040              BIS     #NP.LOOP,R0             ;SET LOOPBACK
9467 063556  052700  000020              BIS     #NP.WRP,R0              ;CLEAR WRITE WRONG PARITY (INVERTED)
9468 063562  004737  065516              JSR     PC,T20WNPR              ;SETUP T20PK2 FOR WRITE NPR
9469 063566  012704  066100              MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9470 063572  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9471 063576  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9472 063602                              FORCERROR       352$                    ;@@DFORCE ERROR IF FORCER=1
9473 063616  103407                      BCS     360$                    ;BR IF CARRY SET (GOOD RETURN)
9474 063620  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9475 063622                              NEXT.ERRNO
9476 063622                      352$:   ERRDF   ERRNO,T204SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063622  104455                                                                      TRAP    C$ERDF
     063624  001623                                                                      .WORD   915
     063626  064727                                                                      .WORD   T204SSR
     063630  012046                                                                      .WORD   PKTSSR
9477 063632  C05237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
9478 063636                      360$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063636  104406                                                                      TRAP    C$CLP1
9479                                    ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
9480 063640  012700  000001              MOV     #1,R0                   ;WRITE 1 BYTE
9481 063644  012701  002312              MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
9482 063650  004737  065556              JSR     PC,T20WFIF              ;SETUP T20PK2 FOR WRITE FIFO
9483 063654  012704  066100              MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9484 063660  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9485 063664  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9486 063670                              FORCERROR       372$                    ;@@DFORCE ERROR IF FORCER=1
9487 063704  103407                      BCS     380$                    ;BR IF CARRY SET (GOOD RETURN)
9488 063706  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9489 063710                              NEXT.ERRNO
9490 063710                      372$:   ERRDF   ERRNO,T205SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063710  104455                                                                      TRAP    C$ERDF
     063712  001624                                                                      .WORD   916
     063714  064772                                                                      .WORD   T205SSR
     063716  012046                                                                      .WORD   PKTSSR
9491 063720  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
9492 063724                      380$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063724  104406                                                                      TRAP    C$CLP1
9493                                    ;       Do a READ FIFO with tape direction OUT to load tape out write latch
9494 063726  012700  000001              MOV     #1,R0                   ;SET READ BYTE COUNT
9495 063732  004737  065536              JSR     PC,T20RFIF              ;SETUP T20PK2 FOR READ FIFO
9496 063736  012704  066100              MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9497 063742  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9498 063746  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9499 063752                              FORCERROR       392$                    ;@@DFORCE ERROR IF FORCER=1
9500 063766  103407                      BCS     400$                    ;BR IF CARRY SET (GOOD RETURN)
9501 063770  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9502 063772                              NEXT.ERRNO
9503 063772                      392$:   ERRDF   ERRNO,T206SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063772  104455                                                                      TRAP    C$ERDF
     063774  001625                                                                      .WORD   917
     063776  065036                                                                      .WORD   T206SSR
     064000  012046                                                                      .WORD   PKTSSR
9504 064002  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
9505 064006                      400$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

TEST 9: READ/WRITE DATA PARITY TEST

```
      064006  104406                                                          TRAP    C$CLP1
 9506                          ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
 9507                          ;               (Read Strobe sets PAR IN H [Parity Error])
 9508 064010  005000                  CLR     R0                      ;IRESV4==>IRSTR = 0
 9509 064012  004737  065612          JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
 9510 064016  012704  066100          MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9511 064022  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9512 064026  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9513 064032                          FORCERROR       412$                    ;@@FORCE ERROR IF FORCEP=1
 9514 064046  103407                  BCS     420$                    ;BR IF CARRY SET (GOOD RETURN)
 9515 064050  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9516 064052                          NEXT.ERRNO
 9517 064052                  412$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      064052  104455                                                          TRAP    C$ERDF
      064054  001626                                                          .WORD   918
      064056  065101                                                          .WORD   T208SSR
      064060  012046                                                          .WORD   PKTSSR
 9518 064062  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
 9519 064066                  420$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      064066  104406                                                          TRAP    C$CLP1
 9520                          ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
 9521 064070  012700  000001          MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
 9522 064074  004737  065612          JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
 9523 064100  012704  066100          MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9524 064104  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9525 064110  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9526 064114                          FORCERROR       432$                    ;@@FORCE ERROR IF FORCER=1
 9527 064130  103407                  BCS     440$                    ;BR IF CARRY SET (GOOD RETURN)
 9528 064132  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9529 064134                          NEXT.ERRNO
 9530 064134                  432$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      064134  104455                                                          TRAP    C$ERDF
      064136  001627                                                          .WORD   919
      064140  065101                                                          .WORD   T208SSR
      064142  012046                                                          .WORD   PKTSSR
 9531 064144  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
 9532 064150                  440$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      064150  104406                                                          TRAP    C$CLP1
 9533
 9534                          ;       Do a Write Subsystem READ STATUS
 9535 064152  004737  065476          JSR     PC,T20SRD               ;SETUP PACKET FOR READ STATUS
 9536 064156  012704  066100          MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9537 064162  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9538 064166  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9539 064172                          FORCERROR       452$                    ;@@FORCE ERROR IF FORCER=1
 9540 064206  103407                  BCS     460$                    ;BR IF CARRY SET (GOOD RETURN)
 9541 064210  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9542 064212                          NEXT.ERRNO
 9543 064212                  452$:   ERRDF   ERRNO,T203SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      064212  104455                                                          TRAP    C$ERDF
      064214  001630                                                          .WORD   920
      064216  064662                                                          .WORD   T203SSR
      064220  012046                                                          .WORD   PKTSSR
 9544 064222  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
 9545 064226                  460$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      064226  104406                                                          TRAP    C$CLP1
 9546                          ;       If Read Data parity error NOT=0  Then Print Error
```

```
TEST  9: READ/WRITE DATA PARITY TEST

 9547 064230  004737  065704                JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 9548 064234  012701  064432                MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
 9549 064240  012702  065772                MOV     #T20BFSTA,R2            ;GET RECV READ STATUS
 9550 064244  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
 9551 064246  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
 9552 064254  042711  100000                BIC     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =0
 9553 064260  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 9554 064262  012701  065752                MOV     #T20BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 9555 064266  012702  064412                MOV     #T20EXP,R2              ;EXPD ADDRESS
 9556 064272  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 9557 064276  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 9558 064302                                FORCERROR       472$,NOTSSR     ;BPO
 9559 064312  103404                        BCS     480$                    ;BR IF YES
 9560 064314                                NEXT.ERRNO
 9561 064314                        472$:   ERRHRD  ERRNO,T20CWP,MSGSTAT    ;REPORT ERROR
      064314  104456                                                                TRAP    C$ERHRD
      064316  001631                                                                .WORD   921
      064320  065357                                                                .WORD   T20CWP
      064322  012350                                                                .WORD   MSGSTAT
 9562 064324                        480$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      064324  104406                                                                TRAP    C$CLP1
 9563
 9564 064326                                FORCEXIT        555$            ;BPO
 9565 064336  013703  002316                MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
 9566 064342  020327  003062                CMP     R3,#TBLEND              ;END OF TSTBLK?
 9567 064346  103002                        BHIS    555$                    ;BR IF YES
 9568 064350  000137  062424                JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
 9569 064354                        555$:
 9570
 9571 064354                                ENDSUB                          ;////////////// END SUBTEST //////////
      064354                                                                        L10074:
      064354  104403                                                                TRAP    C$ESUB
 9572
 9573 064356  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
 9574 064362  001402                        BEQ     560$                    ;BRANCH IF NOT
 9575 064364  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 9576 064370                        560$:
 9577 064370  004737  016456                JSR     PC,TSTLOOP              ;DO ITERATIONS?
 9578 064374  103002                        BCC     565$                    ;BR IF NO
 9579 064376  000137  050056                JMP     T18LOOP                 ;LOOP UNTIL ITERATIONS DONE
 9580 064402
 9581 064402                        565$:   EXIT    TST                     ;////////// EXIT TEST //////////
      064402  104432                                                                TRAP    C$EXIT
      064404  001610                                                                .WORD   L10073-.
 9582
 9583
 9584                               ;+
 9585                               ;LOCAL STORAGE FOR THIS TEST
 9586                               ;-
 9587
 9588
 9589 064406                        T20MSK:                                 ;MASK OF UNTESTED BITS IN READ STATUS
 9590                                                                       ;UNTESTED BITS ARE SET TO 1
 9591 064406      377                       .BYTE   ^C<000>                 ;BYTE 0 MASK
 9592 064407      037                       .BYTE   ^C<340>                 ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
 9593 064410      360                       .BYTE   ^C<017>                 ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
 9594 064411      000                       .BYTE   0                       ;MAKE IT EVEN
```

```
 9595
 9596 064412                            T20EXP:                                 ;BEGIN EXPECTED DATA BUFFER
 9597 064412  000000                            .WORD   0                       ;MESSAGE TYPE
 9598 064414  000000                            .WORD   0                       ;DATA FIELD LENGTH
 9599 064416  000000                            .WORD   0                       ;RBPCR
 9600 064420  000000                            .WORD   0                       ;XST0
 9601 064422  000000                            .WORD   0                       ;XST1
 9602 064424  000000                            .WORD   0                       ;XST2
 9603 064426  000000                            .WORD   0                       ;XST3
 9604 064430  000000                            .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
 9605 064432                            T20EXSTA: .BLKB 64.                     ;EXPECTED READ STATUS AND WRITE FIFO DATA
 9606 064532                            T20EXEND:                                       ;END EXPECTED DATA BUFFER
 9607                                   ;+
 9608                                   ;LOCAL TEXT MESSAGES FOR TEST
 9609                                   ;-
 9610
 9611 064532     122     145     141   TST20ID:        .ASCIZ  'Read/Write Data Parity'
 9612 064561     127     122     111   T20SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
 9613 064616     127     122     111   T202SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
 9614 064662     127     122     111   T203SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
 9615 064727     127     122     111   T204SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
 9616 064772     127     122     111   T205SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
 9617 065036     127     122     111   T206SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
 9618 065101     127     122     111   T208SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Format) Failed'
 9619 065147     122     145     141   T20SWP: .ASCIZ  'Read Data Parity Error (PARERR) Failed to Set after Write Wrong Parity'
 9620 065256     122     145     141   T20RSF: .ASCIZ  'Read Data Parity Error (PARERR) Failed to Clear after RESET FIFO'
 9621 065357     122     145     141   T20CWP: .ASCIZ  'Read Data Parity Error (PARERR) occurred in Data Loopback'
 9622                                           .EVEN
 9623
 9624                                   ;+
 9625                                   ; CLEAR MESSAGE BUFFER
 9626                                   ;-
 9627 065452                            T20CLRBUF:
 9628 065452                                    SAVREG                          ;
                                                                                ;SAVE R1-R5 UNTIL NEXT RETURN
 9629 065456  012701  065752                    MOV     #T20BFR,R1              ;GET MESSAGE BUFFER ADDRESS
 9630 065462  012702  000120                    MOV     #T20BEND-T20BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES
 9631 065466  105021                    10$:    CLRB    (R1)+                   ;CLEAR A BYTE
 9632 065470  005302                            DEC     R2                      ;DONE?
 9633 065472  003375                            BGT     10$                     ;BR IF NO
 9634 065474  000207                            RTS     PC                      ;RETURN
 9635
 9636                                   ;+
 9637                                   ; SETUP T20PK2 PACKET FOR READ STATUS
 9638                                   ;-
 9639 065476                            T20SRD:
 9640 065476  004737  065452                    JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
 9641 065502  012700  066110                    MOV     #T20DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
 9642 065506  112720  000005                    MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
 9643 065512  105010                            CLRB    (R0)                    ;CLEAR BSEL1
 9644 065514  000207                            RTS     PC                      ;RETURN
 9645
 9646
 9647                                   ;+
 9648                                   ; SETUP T20PK2 PACKET FOR WRITE NPR
 9649                                   ;
 9650                                   ; INPUT:
 9651                                   ;       R0 CONTAINS BSEL1 NPR DATA
```

TEST 9: READ/WRITE DATA PARITY TEST

```
                                        ;
                                        ;-
065516                                  T20WNPR:
065516  004737  065452                        JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
065522  012701  066110                        MOV     @T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
065526  112721  000011                        MOVB    @PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
065532  110011                              MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
065534  000207                              RTS     PC                      ;RETURN
                                        ;+
                                        ; SETUP T20PK2 PACKET FOR READ FIFO
                                        ;
                                        ; INPUT:
                                        ;       R0 CONTAINS SEL2 BYTE COUNT
                                        ;-
065536                                  T20RFIF:
065536  004737  065452                        JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
065542  012701  066110                        MOV     @T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
065546  112721  000003                        MOVB    @PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
065552  110021                              MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
065554  000207                              RTS     PC                      ;RETURN
                                        ;+
                                        ; SETUP T20PK2 PACKET FOR WRITE FIFO
                                        ;
                                        ; INPUT:
                                        ;       R0 CONTAINS BYTE COUNT
                                        ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
                                        ;-
065556                                  T20WFIF:
065556                                      SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
065562  004737  065452                        JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
065566  012702  066110                        MOV     @T20DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
065572  112722  000004                        MOVB    @PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
065576  110022                              MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
065600  005022                              CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
065602  112122                          10$:MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
065604  005300                              DEC     R0                      ;DONE ALL BYTES?
065606  003375                              BGT     10$                     ;BR IF NO
065610  000207                              RTS     PC                      ;RETURN
                                        ;+
                                        ; SETUP T20PK2 FOR WRITE FORMAT TRANSPORT REGISTER
                                        ;
                                        ; INPUT:
                                        ;       R0 CONTAINS DRIVING DATA PATTERN
                                        ;-
065612                                  T20WFMT:
065612  004737  0o5452                        JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
065616  012701  066110                        MOV     @T20DT2,R1              ;WRITE SUBSYSTEM DA~. BUFFER
065622  112721  000007                        MOVB    @PW.WFMT,(R1)+          ;STORE WRITE FORMAT IN BSEL0
065626  110021                              MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
065630  000207                              RTS     PC                      ;RETURN
                                        ;+
                                        ; SETUP T20PK2 PACKET FOR WRITE MISC.
                                        ;
                                        ;       R0 CONTAINS WRITE MISC DATA
                                        ;-
```

TEST 9: READ/WRITE DATA PARITY TEST

```
 9709 065632                            T20WMISC:
 9710 065632  012701  066110                MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
 9711 065636  112721  000010                MOVB    #PW.WMISC,(R1)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
 9712 065642  110011                        MOVB    R0,(R1)                 ;STORE INVERT EXTENDED FEATURES IN BSEL1
 9713 065644  000207                        RTS     PC                      ;RETURN
 9714                                   ;+
 9715                                   ; SETUP T20PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
 9716                                   ;-
 9717 065646                            T20SEXT:
 9718 065646  012700  066110                MOV     #T20DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
 9719 065652  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
 9720 065656  112710  000200                MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
 9721 065662  000207                        RTS     PC                      ;RETURN
 9722                                   ;+
 9723                                   ; CLEAR EXPECTED DATA MESSAGE BUFFER
 9724                                   ;-
 9725 065664                            T20CLEXP:
 9726 065664  012701  064412                MOV     #T20EXP,R1              ;GET EXPD ADDRESS
 9727 065670  012700  000120                MOV     #T20EXEND-T20EXP,R0     ;GET EXPD SIZE
 9728 065674  105021                    10$:    CLRB    (R1)+                   ;CLEAR A BYTE
 9729 065676  005300                        DEC     R0                      ;DONE?
 9730 065700  003375                        BGT     10$                     ;BR IF NO
 9731 065702  000207                        RTS     PC                      ;RETURN
 9732
 9733                                   ;+
 9734                                   ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
 9735                                   ;-
 9736 065704                            T20SETEXP:
 9737 065704  012702  064412                MOV     #T20EXP,R2              ;GET EXPD
 9738 065710  012703  065752                MOV     #T20BFR,R3              ;GET READ STATUS RECV BUFFER
 9739 065714  012700  000010                MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
 9740 065720  012322                    5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
 9741 065722  005300                        DEC     R0                      ;DONE WORDS 0-7 WORDS?
 9742 065724  003375                        BGT     5$                      ;BR IF NO
 9743 065726  000207                        RTS     PC                      ;RETURN
 9744
 9745
 9746
 9750                                   ;
 9751                                   ;WRITE CHARACTERISTICS COMMAND PACKET
 9752                                   ;
 9753 065730                            T20PACKET:                              ;COMMAND PACKET FOR TEST
 9754 065730  100004                        .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
 9755 065732  065740                        .WORD   T20DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
 9756 065734  000000                        .WORD   0
 9757 065736  000012                        .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 9758
 9759 065740                            T20DATA:                                ;CHARACTERISTICS DATA BLOCK
 9760 065740  065752                        .WORD   T20BFR                  ;ADDRESS OF MESSAGE BUFFER
 9761 065742  000000                        .WORD   0
 9762 065744  000024                        .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
 9763 065746  000000                        .WORD   0                       ;ESS,ENB,EAI,ERI
 9764 065750  000007                        .WORD   7                       ;EXTENDED FEATURES UNIT NO.
 9765
 9766
 9767                                   ;MESSAGE BUFFER FOR ALL TEST 17 COMMANDS
 9768
```

TEST 9: READ/WRITE DATA PARITY TEST

```
 9769 065752                    T20BFR:                                 ;BEGIN MESSAGE BUFFER
 9770 065752  000000                    .WORD   0                       ;MESSAGE TYPE
 9771 065754  000000                    .WORD   0                       ;DATA FIELD LENGTH
 9772 065756  000000                    .WORD   0                       ;RBPCR
 9773 065760  000000                    .WORD   0                       ;XST0
 9774 065762  000000                    .WORD   0                       ;XST1
 9775 065764  000000                    .WORD   0                       ;XST2
 9776 065766  000000                    .WORD   0                       ;XST3
 9777 065770  000000                    .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
 9778 065772                    T20BFSTA: .BLKB 64.                     ;READ STATUS AND WRITE FIFO BUFFER
 9779 066072                    T20BEND:                                ;END OF MESSAGE BUFFER
 9780                           ;
 9781                           ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
 9782                           ;
 9784         066100                    .=<.+10>&177770
 9786 066100                    T20PK2:
 9787 066100  100006                    .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
 9788 066102  066110                    .WORD   T20DT2                  ;LOW ADDRESS OF DATA BLOCK
 9789 066104  C00000                    .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
 9790 066106  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 9791
 9792 066110                    T20DT2:                                 ;DATA BLOCK
 9793 066110     000                    .BYTE   0                       ;BSEL0
 9794 066111     000                    .BYTE   0                       ;BSEL1
 9795 066112  000000                    .WORD   0                       ;SEL2
 9796 066114                            .BLKB   64.                     ;WRITE FIFO DATA OUTPUT BUFFER
 9797
 9798
 9799 066214                            ENDTST
      066214                                                            L10073:
      066214  104401                                                           TRAP    C$ETST
 9800                                   .SBTTL  TEST 10: MANUAL INTERVENTION
 9801
 9802                           ;THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")
 9803                           ;THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
 9804                           ;THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
 9805                           ;THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
 9806                           ;SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
 9807                           ;THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
 9808                           ;ARE BY TYPING <CTRL-C> OR SELECTING CODE 7.
 9809                           ;SELECTION CODES AND SUBROUTINES ARE:
 9810                           ;
 9811                           ;
 9812                           ;       CODE    ROUTINE
 9813                           ;
 9814                           ;         0     HELP. PRINTS THIS MENU.
 9815                           ;         1     TURN ON ALL M7196 LED INDICATORS
 9816                           ;         2     TURN OFF ALL M7196 LED INDICATORS
 9817                           ;         3     OFFLINE/ONLINE ATTENTION TEST
 9818                           ;         4     WRITE-PROTECT TEST
 9819                           ;         5     INITIATE TRANSPORT SERVO EXERCISER
 9820                           ;         6     PRINT EXTENDED TRANSPORT STATUS
 9821                           ;         7     EXIT (RETURN TO SUPERVISOR)
 9822                           ;
 9823                           ;
 9824                           ;EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:
 9825                           ;
```

```
9826                                    ;
9827                                    ;
9828                                    ;PRINTS OUT THE MENU ON THE CONSOLE TERMINAL.
9829                                    ;
9830                                    ;
9831                                    ;CAUSES ALL THREE LED INDICATORS ON THE M7196 MODULE
9832                                    ;TO BE ILLUMINATED. AFTER INITIATING THIS ROUTINE, THE OPERATOR
9833                                    ;SHOULD OBSERVE THE LED'S AND VERIFY THAT THEY ARE INDEED ALL LIT.
9834                                    ;THIS ROUTINE FIRST USES THE WRITE SUBSYSTEM MEMORY COMMAND TO
9835                                    ;SET THE FORCE WRONG PARITY FLIP-FLOP, WHICH SERVES TO DRIVE THE
9836                                    ;"PROCESSOR NOT OK" LED. THEN IT ENTERS A LOOP THAT CONTINUALLY
9837                                    ;WRITES THE LOW BYTE OF TSDB AND READS THE TSSR. THESE LATTER TWO
9838                                    ;OPERATIONS WILL CAUSE THE "NOT SSR" AND "DRIVING BUS" LED'S TO
9839                                    ;GLOW -- THEY ARE NOT REALLY LIT AT ALL TIMES BUT SHOULD APPEAR
9840                                    ;REASONABLY VISIBLE.
9841                                    ;
9842                                    ;
9843                                    ;INITIALIZES THE CONTROLLER TO CAUSE ALL LED'S TO
9844                                    ;EXTINGUISH.
9845                                    ;
9846                                    ;
9847                                    ;
9848                                    ;
9849                                    ;THIS ROUTINE INITIALIZES THE CONTROLLER, ISSUES A
9850                                    ;WRITE CHARACTERISTICS COMMAND TO ENABLE ATTENTION INTERRUPTS,
9851                                    ;ISSUES A MESSAGE BUFFER RELEASE COMMAND, PRINTS A MESSAGE ON THE
9852                                    ;CONSOLE TERMINAL INSTRUCTING THE OPERATOR TO TOGGLE THE ON-LINE
9853                                    ;SWITCH ON THE TRANSPORT, THEN WAITS FOR AN ATTENTION INTERRUPT.
9854                                    ;EACH TIME THE TRANSPORT TRANSITIONS FROM ON-LINE TO OFF-LINE OR
9855                                    ;VICE-VERSA, AN ATTENTION INTERRUPT SHOULD BE GENERATED. THE PROGRAM
9856                                    ;WILL REPORT THE INTERRUPT AND THE CURRENT STATE OF THE TRANSPORT.
9857                                    ;THE OPERATOR SHOULD VERIFY THAT THE REPORTED STATE MATCHES THE
9858                                    ;STATE INDICATED BY THE LED ON THE FRONT PANEL OF THE TRANSPORT.
9859                                    ;IN ADDITION, WHEN THE TRANSPORT IS PLACED OFF-LINE, THE PROGRAM
9860                                    ;ISSUES A SEQUENCE OF TAPE-MOTION COMMANDS (READ, WRITE, POSITION, ETC.
9861                                    ;AND VERIFIES THAT, FOR EACH COMMAND, FUNCTION REJECT TERMINATION
9862                                    ;RESULTS, ALONG WITH THE NON-EXECUTABLE FUNCTION (NEF) ERROR BIT BEING
9863                                    ;SET.
9864                                    ;
9865                                    ;THIS ROUTINE INSTRUCTS THE OPERATOR TO MOUNT A SCRATCH
9866                                    ;TAPE REEL THAT DOES NOT HAVE A WRITE-ENABLE RING INSTALLED, THEN
9867                                    ;WAITS FOR THE OPERATOR TO RESPOND THAT THIS HAS BEEN ACCOMPLISHED.
9868                                    ;UPON THE RESPONSE, THE PROGRAM VERIFIES THAT THE TRANSPORT SHOWS
9869                                    ;A WRITE-PROTECTED STATUS, THEN ATTEMPTS TO WRITE DATA ON THE
9870                                    ;TAPE AND EXPECTS THE APPROPRIATE ERROR TERMINATION INDICATING THAT
9871                                    ;THE WRITE FUNCTION COULD NOT BE PERFORMED BECAUSE THE REEL IS
9872                                    ;WRITE-PROTECTED.  IF THE APPROPRIATE TERMINATION IS NOT RECEIVED,
9873                                    ;AN ERROR IS REPORTED.
9874                                    ;
9875                                    ;
9876                                    ;
9877                                    ;INSTRUCTS THE OPERATOR TO PLACE THE TAPE TRANSPORT(S)
9878                                    ;ON-LINE (IF ANY ARE OFF-LINE) THEN ATTEMPTS TO PERFORM AN EXTENDED
9879                                    ;STATUS READOUT.  FOR EACH TRANSPORT EQUIPPED WITH THIS FEATURE,
9880                                    ;THE PROGRAM FORMATS AND PRINTS OUT THE RESULTING STATUS. IF THE
9881                                    ;TRANSPORT IS NOT EQUIPPED WITH THIS FEATURE, A MESSAGE INDICATING
9882                                    ;SUCH IS ISSUED.
```

TEST 10: MANUAL INTERVENTION

```
 9883                                   ;
 9884                                   ;
 9885                                   ;
 9886                                   ;
 9887 066216                            BGNTST
      066216                                                                  T10::
 9892 066216                            RFLAGS  R0              ;GET OPERATOR FLAGS
      066216  104421                                                         TRAP    C$RFLA
 9893 066220  001403                    BEQ     21$             ;BR, IF OK TO RUN
 9894 066222  012700  071570            MOV     @T38NE,R0       ;"TEST NOT EXECUTED"
 9895 066226  000402                    BR      3$              ;JUMP IF NOT FIRST TEST
 9896 066230                            21$:
 9897 066230  012700  072705            MOV     @T38ID,R0       ;TEST ID MESSAGE
 9898 066234  004737  016510            3$:     JSR     PC,TSTSETUP     ;DO THE COMMON SETUP
 9899 066240  004737  017762            JSR     PC,CHKMAN       ;IS MANUAL INTERVENTION ALLOWED?
 9900 066244  103402                    BCS     22$             ;BR, IF MANUAL INTER ALLOWED
 9901 066246  000137  070772            JMP     64$             ;JUMP IF NOT ALLOWED
 9902 066252                            22$:
 9906 066252  C05037  002222            2$:     CLR     FATFLG          ;CLEAR THE FATAL ERROR FLAG
 9907 066256  012737  176750  071004    MOV     @65000.,T38DLY  ;SET UP DELAY COUNTER
 9908 066264  004737  015774            5$:     JSR     PC,SOFINIT      ;DO A SOFT INIT
 9909 066270  103427                    BCS     23$             ;BRANCH IF OK
 9910 066272  010001                    MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
 9911 066274  032701  000200            BIT     @SSR,R1         ;CHECK FOR TSSR SET
 9912 066300  001023                    BNE     23$             ;KEEP GOING IF NOT SET
 9913 066302                            DELAY   250             ;CALL DELAY ROUTINE
      066302  012727  000250                                                   MOV     @250,(PC)+
      066306  000000                                                          .WORD   0
      066310  013727  002116                                                  MOV     L$DLY,(PC)+
      066314  000000                                                          .WORD   0
      066316  005367  177772                                                  DEC     -6(PC)
      066322  001375                                                          BNE     .-4
      066324  005367  177756                                                  DEC     -22(PC)
      066330  001367                                                          BNE     .-20
 9914 066332  005337  071004            DEC     T38DLY          ;BUMP COUNTER DOWN
 9915 066336  001352                    BNE     5$              ;BR, IF MORE TIME LEFT
 9916 066340                            ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
      066340  104455                                                         TRAP    C$ERDF
      066342  001751                                                        .WORD   1001
      066344  003652                                                        .WORD   SFIERR
      066346  012034                                                        .WORD   SFIMSG
 9917 066350  012700  072732    23$:    MOV     @MIMENU,R0      ;MENU OF MANUAL INTERVENTIONS
 9918 066354  012701  000005            MOV     @5,R1           ;MAXIMUM ALLOWED SELECTION
 9919 066360  004737  017540            JSR     PC,GETSEL       ;GO GET THE OPERATORS SELECTION
 9920 066364  010004                    MOV     R0,R4           ;GET NUMBER FROM ROUTINE
 9921 066366  006304                    ASL     R4              ;CONVERT TO WORD OFFSET
 9922 066370  000174  066374            JMP     @6$(R4)         ;JUMP TO PROPER LOOP
 9923 066374  066252            6$:     .WORD   2$              ;RETYPE THE MENU
 9924 066376  066410                    .WORD   10$             ; 1 TURN ON LED'S
 9925 066400  066672                    .WORD   15$             ; 2 TURN OFF LED'S
 9926 066402  067124                    .WORD   20$             ; 3 ONLINE ATTENTION
 9927 066404  067560                    .WORD   25$             ; 4 WRITE PROTECT
 9928 066406  070514                    .WORD   35$             ; 5 EXTENDED TRANSPORT STATUS
 9929 066410                            10$:    PRINTF  @T38MS2         ;TELL OPERATOR TO CNTRL-C FOR EXIT
      066410  012746  072601                                                 MOV     @T38MS2,-(SP)
      066414  012746  000001                                                 MOV     @1,-(SP)
      066420  010600                                                         MOV     SP,R0
```

TEST 10: MANUAL INTERVENTION

```
        066422  104417                                                          TRAP    C$PNTF
        066424  062706  000004                                                  ADD     #4,SP
 9930   066430  004737  073342          JSR     PC,T38REST      ;SET PACKET TO INITIAL VALUES
 9931   066434  004737  015774          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
 9932   066440  103405                  BCS     100$            ;BR IF SOFT INIT = OK
 9936   066442  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
 9937   066444                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
        066444  104455                                                          TRAP    C$ERDF
        066446  001752                                                          .WORD   1002
        066450  003652                                                          .WORD   SFIERR
        066452  012034                                                          .WORD   SFIMSG
 9938   066454  013737  002202  071530  100$:   MOV     UNITN,T38DSW    ;SET UNIT NUMBER
 9939
 9940   066462  012704  071510          MOV     #T38PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
 9941   066466  004737  010662          JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
 9942   066472  103405                  BCS     110$            ;BR, IF COMMAND ISSUED OK
 9946   066474  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
 9947   066476                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
        066476  104456                                                          TRAP    C$ERHRD
        066500  001753                                                          .WORD   1003
        066502  005056                                                          .WORD   WRTMSG
        066504  012034                                                          .WORD   SFIMSG
 9948   066506                  110$:
 9949   066506  112737  000000  071021          MOVB    #0,T38BS1       ;CLEAR BIT #4
 9950   066514  112737  000011  071020          MOVB    #11,T38BS0      ;WRITE MISC COMMAND
 9951   066522  012704  071010          MOV     #T38PACKET,R4   ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 9952                                  ;
 9953                                  ;NOTE: THIS COMMAND TURNS ON THE PROCESSOR FAIL LED
 9954                                  ;
 9955   066526  010465  000000          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
 9956   066532  004737  016336          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
 9957   066536  103405                  BCS     150$            ;BR IF CARRY SET (GOOD RETURN)
 9958   066540  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
 9962   066542                          ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        066542  104455                                                          TRAP    C$ERDF
        066544  001754                                                          .WORD   1004
        066546  072206                                                          .WORD   T38SSR
        066550  012046                                                          .WORD   PKTSSR
 9963   066552                  150$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        066552  104406                                                          TRAP    C$CLP1
 9964   066554                          SETPRI  #PRI07          ;RAISE THE PRIORITY
        066554  012700  000340                                                  MOV     #PRI07,R0
        066560  104441                                                          TRAP    C$SPRI
 9965   066562  005037  070776          CLR     TTION2          ;ASSUME INTERRUPTS ARE ENABLED
 9966   066566  032737  000100  177560  BIT     #100,@#TTICSR   ;ARE TTI INTERRUPTS ON ?
 9967   066574  001005                  BNE     701$            ;BRANCH IF YES
 9968   066576  005237  070776          INC     TTION2          ;FLAG SET IF INTERRUPTS OFF
 9969   066602  052737  000100  177560  BIS     #100,@#TTICSR   ;ENABLE INTERRUPTS
 9970   066610  012701  000060  701$:   MOV     #TTIVEC,R1      ;START OF TTI VECTORS
 9971   066614  011137  071000          MOV     (R1),TVSAV2     ;SAVE THE CURRENT TTI VECTOR
 9972   066620  012721  070276          MOV     #590$,(R1)+     ;SET NEW INTERRUPT ROUTINE
 9973   066624  011137  071002          MOV     (R1),TPSAV2     ;SAVE THE VECTOR PRIORITY
 9974   066630  012711  000340          MOV     #PRI07,(R1)     ;USE PRIORITY SEVEN
 9975   066634                          SETPRI  #PRI00          ;LOWER INTERRUPT BR LEVEL
        066634  012700  000000                                                  MOV     #PRI00,R0
        066640  104441                                                          TRAP    C$SPRI
 9976   066642  012701  177777          MOV     #-1,R1          ;DATA TO WRITE TO TSDB
```

TEST 10: MANUAL INTERVENTION

```
 9977 066646  000240                  12$:    NOP                         ;ALLOW OPERATOR TO TYPE ^C
 9978 066650  012702  001750                   MOV     #1000.,R2           ;SET-UP INNER LOOP
 9979 066654  110165  000000           14$:    MOVB    R1,TSDB(R5)         ;WRITE DATA TO TSDB
 9980 066660  016500  000002                   MOV     TSSR(R5),R0         ;READ TSSR
 9981 066664  005302                          DEC     R2                  ;REDUCE INNER COUNT
 9982 066666  001372                          BNE     14$                 ;LOOP TILL EXPIRES
 9983 066670  000766                          BR      12$                 ;LOOP UNTIL HALTED
 9984
 9985 066672                           15$:    PRINTF  #T38MS2             ;TYPE CNTL C TO EXIT
      066672  012746  072601                                                       MOV     #T38MS2,-(SP)
      066676  012746  000001                                                       MOV     #1,-(SP)
      066702  010600                                                               MOV     SP,R0
      066704  104417                                                               TRAP    C$PNTF
      066706  062706  000004                                                       ADD     #4,SP
 9986 066712  004737  015774                   JSR     PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
 9987 066716  103405                          BCS     200$                ;BR IF SOFT INIT = OK
 9991 066720  010001                          MOV     R0,R1               ;SAVE CONTENTS OF TSSR
 9992 066722                                  ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      066722  104455                                                               TRAP    C$ERDF
      066724  001755                                                               .WORD   1005
      066726  003652                                                               .WORD   SFIERR
      066730  012034                                                               .WORD   SFIMSG
 9993 066732                           200$:
 9994 066732  013737  002202  071530       MOV     UNITN,T38DSW        ;SET UNIT NUMBER
 9995 066740  012704  071510               MOV     #T38PK2,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
 9996 066744  004737  010662               JSR     PC,WRTCHR           ;ISSUE WRITE CHARACTERISTICS
 9997 066750  103405                       BCS     210$                ;BR, IF COMMAND ISSUED OK
10001 066752  010001                       MOV     R0,R1               ;SAVE CONTENTS OF TSSR
10002 066754                               ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
      066754  104456                                                               TRAP    C$ERHRD
      066756  001756                                                               .WORD   1006
      066760  005056                                                               .WORD   WRTMSG
      066762  012034                                                               .WORD   SFIMSG
10003                                 ;*****************************************************************
10004                                 ;
10005                                 ;       THIS WRITE SUB-SYSTEM MEMORY COMMAND JUST HOLDS THE LEDS OFF
10006                                 ;
10007                                 ;*****************************************************************
10008 066764                          210$:
10009 066764  112737  000000  071021       MOVB    #0,T38BS1           ;CLEAR BIT #4
10010 066772  112737  000025  071020       MOVB    #25,T38BS0          ;STOP DRIVE TEST 22
10011 067000  012704  071010               MOV     #T38PACKET,R4       ;SET UP NEW WRT. SUBSYS MEM. COMMAND
10012 067004  010465  000000               MOV     R4,TSDB(R5)         ;SET THE PACKET ADDRESS
10013 067010  004737  016336               JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
10014 067014  103405                       BCS     250$                ;BR IF CARRY SET (GOOD RETURN)
10015 067016  010001                       MOV     R0,R1               ;SAVE CONTENTS OF TSSR
10019 067020                               ERRDF   ERRNO,T38SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      067020  104455                                                               TRAP    C$ERDF
      067022  001757                                                               .WORD   1007
      067024  072206                                                               .WORD   T38SSR
      067026  012046                                                               .WORD   PKTSSR
10020 067030                          250$:   CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
      067030  104406                                                               TRAP    C$CLP1
10021 067032                               SETPRI  #PRI07              ;RAISE THE PRIORITY
      067032  012700  000340                                                       MOV     #PRI07,R0
      067036  104441                                                               TRAP    C$SPRI
10022 067040  005037  070776               CLR     TTION2              ;ASSUME INTERRUPTS ARE ENABLED
```

TEST 10: MANUAL INTERVENTION

```
10023 067044  032737  000100  177560          BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
10024 067052  001005                          BNE     710$                    ;BRANCH IF YES
10025 067054  005237  070776                  INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
10026 067060  052737  000100  177560          BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
10027 067066  012701  000060          710$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
10028 067072  011137  071000                  MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
10029 067076  012721  070276                  MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
10030 067102  011137  071002                  MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
10031 067106  012711  000340                  MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
10032 067112                                  SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
      067112  012700  000000                                                          MOV     #PRI00,R0
      067116  104441                                                                  TRAP    C$SPRI
10033 067120  000240          260$:   NOP                                     ;ALLOW CNTL C
10034 067122  000776                  BR      260$                            ;LOOP UNTIL STOPPED
10035
10036
10037 067124                  20$:    PRINTF  #T38MS2                         ;TELL'EM WHAT TO TYPE
      067124  012746  072601                                                          MOV     #T38MS2,-(SP)
      067130  012746  000001                                                          MOV     #1,-(SP)
      067134  010600                                                                  MOV     SP,R0
      067136  104417                                                                  TRAP    C$PNTF
      067140  062706  000004                                                          ADD     #4,SP
10038 067144                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      067144  012700  000000                                                          MOV     #PRI00,R0
      067150  104441                                                                  TRAP    C$SPRI
10039 067152  005037  002224                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
10040 067156  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
10041 067162  103405                          BCS     300$                    ;BR IF SOFT INIT = OK
10045 067164  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10046 067166                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      067166  104455                                                                  TRAP    C$ERDF
      067170  001760                                                                  .WORD   1008
      067172  003652                                                                  .WORD   SFIERR
      067174  012034                                                                  .WORD   SFIMSG
10047 067176                  300$:
10048 067176  013737  002202  071530          MOV     UNITN,T38DSW            ;SET UNIT NUMBER IN PACKET
10049 067204  012737  000040  071526          MOV     #BIT5,T38EAI            ;ENABLE ATTENTION INTERRUPTS
10050 067212  012704  071510                  MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10051 067216  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10052 067222  103405                          BCS     310$                    ;BR, IF COMMAND ISSUED OK
10056 067224  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10057 067226                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      067226  104456                                                                  TRAP    C$ERHRD
      067230  001761                                                                  .WORD   1009
      067232  005056                                                                  .WORD   WRTMSG
      067234  012034                                                                  .WORD   SFIMSG
10058 067236                  310$:
10059 067236  012704  01540                   MOV     #T38PK3,R4              ;SET UP NEW PACKET FOR MESS BUF REL
10060 067242  010465  000000                  MOV     R4,TSDB(R5)             ;MESSAGE BUFFER RELEASE,ACK,CVC=1 CMD
10061 067246  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
10062 067252  005002                          CLR     R2                      ;MAKE SURE ALL IS CLEAR
10063 067254  016501  000002                  MOV     TSSR(R5),R1             ;GET TSSR STATUS
10064 067260  032701  000100                  BIT     #OFL,R1                 ;IS OFL SET
10065 067264  001402                          BEQ     320$                    ;BR, IF OFL IS NOT SET
10066 067266  052702  000100                  BIS     #OFL,R2                 ;SET OFL IN EXPECTED
10067 067272  052702  000200          320$:   BIS     #SSR,R2                 ;SET UP EXPECTED
10068 067276  020201                          CMP     R2,R1                   ;IS EVERYTHING OK
```

TEST 10: MANUAL INTERVENTION

```
10069 067300  001404                        BEQ     350$                ;BR, IF ALL IS WELL
10073 067302                                ERRHRD  ERRNO,T38SST,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      067302  104456                                                    TRAP    C$ERHRD
      067304  001762                                                    .WORD   1010
      067306  072416                                                    .WORD   T38SST
      067310  012046                                                    .WORD   PKTSSR
10074 067312                        350$:   CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
      067312  104406                                                    TRAP    C$CLP1
10075 067314                                PRINTF  @T38MS1             ;TELL OPERATOR TO TOGGLE SWITCH
      067314  012746  072506                                            MOV     @T38MS1,-(SP)
      067320  012746  000001                                            MOV     #1,-(SP)
      067324  010600                                                    MOV     SP,R0
      067326  104417                                                    TRAP    C$PNTF
      067330  062706  000004                                            ADD     #4,SP
10076 067334                                PRINTF  @T38MS2             ;TELL OPERATOR TO DO ^C TO EXIT
      067334  012746  072601                                            MOV     @T38MS2,-(SP)
      067340  012746  000001                                            MOV     #1,-(SP)
      067344  010600                                                    MOV     SP,R0
      067346  104417                                                    TRAP    C$PNTF
      067350  062706  000004                                            ADD     #4,SP
10077 067354                                SETPRI  @PRI07              ;RAISE THE PRIORITY
      067354  012700  000340                                            MOV     @PRI07,R0
      067360  104441                                                    TRAP    C$SPRI
10078 067362  005037  070776                CLR     TTION2              ;ASSUME INTERRUPTS ARE ENABLED
10079 067366  032737  000100  177560        BIT     #100,@#TTICSR       ;ARE TTI INTERRUPTS ON ?
10080 067374  001005                        BNE     720$                ;BRANCH IF YES
10081 067376  005237  070776                INC     TTION2              ;FLAG SET IF INTERRUPTS OFF
10082 067402  052737  000100  177560        BIS     #100,@#TTICSR       ;ENABLE INTERRUPTS
10083 067410  012701  000060        720$:   MOV     @TTIVEC,R1          ;START OF TTI VECTORS
10084 067414  011137  071000                MOV     (R1),TVSAV2         ;SAVE THE CURRENT TTI VECTOR
10085 067420  012721  070276                MOV     #590$,(R1)+         ;SET NEW INTERRUPT ROUTINE
10086 067424  011137  071002                MOV     (R1),TPSAV2         ;SAVE THE VECTOR PRIORITY
10087 067430  012711  000340                MOV     #PRI07,(R1)         ;USE PRIORITY SEVEN
10088 067434                                SETPRI  #PRI00              ;LOWER INTERRUPT BR LEVEL
      067434  012700  000000                                            MOV     #PRI00,R0
      067440  104441                                                    TRAP    C$SPRI
10089 067442  000240                360$:   NOP                         ;ALLOW CONTROL C
10090 067444  005737  002224                TST     INTRECV             ;DID AN INTERRUPT OCCUR ?
10091 067450  001001                        BNE     370$                ;BRANCH IF YES
10092 067452  000773                        BR      360$                ;WAIT SOME MORE FOR INTERRUPT
10093 067454                        370$:   PRINTF  @T38INT             ;"INTERRUPT RECEIVED"
      067454  012746  072276                                            MOV     @T38INT,-(SP)
      067460  012746  000001                                            MOV     #1,-(SP)
      067464  010600                                                    MOV     SP,R0
      067466  104417                                                    TRAP    C$PNTF
      067470  062706  000004                                            ADD     #4,SP
10094 067474  016501  000002                MOV     TSSR(R5),R1         ;READ TSSR STATUS
10095 067500  032701  000100                BIT     #OFL,R1             ;CHECK THE OFF-LINE BIT
10096 067504  001011                        BNE     380$                ;BR, IF DRIVE IS OFF-LINE
10097 067506                                PRINTF  @T38ONL             ;"DRIVE IS NOW ON-LINE"
      067506  012746  072326                                            MOV     @T38ONL,-(SP)
      067512  012746  000001                                            MOV     #1,-(SP)
      067516  010600                                                    MOV     SP,R0
      067520  104417                                                    TRAP    C$PNTF
      067522  062706  000004                                            ADD     #4,SP
10098 067526  000410                        BR      390$                ;ALMOST DONE
10099 067530                        380$:   PRINTF  @T38OFL             ;"DRIVE IS NOW OFF-LINE"
```

TEST 10: MANUAL INTERVENTION

```
        067530  012746  072362                                                          MOV     #T38OFL,-(SP)
        067534  012746  000001                                                          MOV     #1,-(SP)
        067540  010600                                                                  MOV     SP,R0
        067542  104417                                                                  TRAP    C$PNTF
        067544  062706  000004                                                          ADD     #4,SP
 10100  067550  005037  002224          390$:   CLR     INTRECV                 ;CLEAR INTERRUPT FLAG
 10101  067554  000137  067176                  JMP     300$                    ;TRY AGAIN
 10102  067560                          25$:    GMANIL  T38MSG,T38DAT,-1,NO     ;WAIT FOR OPERATOR TO MOUNT TAPE
        067560  104443                                                                  TRAP    C$GMAN
        067562  000404                                                                  BR      10000$
        067564  073340                                                                  .WORD   T38DAT
        067566  000120                                                                  .WORD   T$CODE
        067570  072645                                                                  .WORD   T38MSG
        067572  177777                                                                  .WORD   -1
        067574                                                                  10000$:
 10103  067574                                  BNCOMPLETE 25$                  ;RETRY IF ERROR
        067574  103371                                                                  BCC     25$
 10104  067576  005737  073340                  TST     T38DAT                  ;DID OPERATOR SAY 'YES' ?
 10105  067602  C01002                          BNE     27$                     ;BRANCH IF YES
 10106  067604  000137  066252                  JMP     2$                      ;RETURN TO MAIN MENU
 10107  067610                          27$:
 10108  067610  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 10109  067614  103405                          BCS     400$                    ;BR IF SOFT INIT = OK
 10113  067616  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 10114  067620                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
        067620  104455                                                                  TRAP    C$ERDF
        067622  001763                                                                  .WORD   1011
        067624  003652                                                                  .WORD   SFIERR
        067626  012034                                                                  .WORD   SFIMSG
 10115  067630                          400$:   CKLOOP                          ;LOOP IF SELECTED
        067630  104406                                                                  TRAP    C$CLP1
 10116  067632  013737  002202  071530          MOV     UNITN,T38DSW            ;SET UNIT NUMBER
 10117  067640  012704  071510                  MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 10118  067644  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 10119  067650  103405                          BCS     410$                    ;BR, IF COMMAND ISSUED OK
 10123  067652  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 10124  067654                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
        067654  104456                                                                  TRAP    C$ERHRD
        067656  001764                                                                  .WORD   1012
        067660  005056                                                                  .WORD   WRTMSG
        067662  012034                                                                  .WORD   SFIMSG
 10125  067664                          410$:   CKLOOP                          ;LOOP IF SELECTED
        067664  104406                                                                  TRAP    C$CLP1
 10126  067666  013701  071034                  MOV     T38BFR+6,R1             ;PICK UP XSTO CONTENTS
 10127  067672  010102                          MOV     R1,R2                   ;SET UP EXPECTED
 10128  067674  052702  000004                  BIS     #BIT2,R2                ;SET UP THE WRITE LOCKED BIT
 10129  067700  020102                          CMP     R1,R2                   ;ARE THEY CORRECT
 10130  067702  001406                          BEQ     430$                    ;BR, IF ALL IS WELL (OK)
 10134  067704                                  ERRHRD  ERRNO,T38WRL,EXPREC     ;"WRITE LOCKED BIT IS NOT SET ETC."
        067704  104456                                                                  TRAP    C$ERHRD
        067706  001765                                                                  .WORD   1013
        067710  072024                                                                  .WORD   T38WRL
        067712  015474                                                                  .WORD   EXPREC
 10135  067714  005237  002222                  INC     FATFLG                  ;SET FATAL FLAG
 10136  067720                          430$:   CKLOOP                          ;LOOP IF SELECTED
        067720  104406                                                                  TRAP    C$CLP1
 10137  067722  005737  002222                  TST     FATFLG                  ;WAS THE DRIVE NOT WRITE LOCKED
```

TEST 10: MANUAL INTERVENTION

```
 10138 067726  001402                        BEQ     435$                    ;BR, IF FLAG NOT SET
 10139 067730  000137  066252                JMP     2$                      ;RE-WRITE MENU
 10140 067734  017737  113164  071562  435$: MOV     @FREE,T38WR             ;SET UP WRITE BUFFER ADDRESS
 10141 067742  012704  071560                MOV     #T38PK4,R4              ;GET PACKET ADDRESS
 10142 067746  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 10143 067752  004737  016250                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 10144 067756  016501  000002                MOV     TSSR(R5),R1             ;GET TSSR
 10145 067762  012702  100206                MOV     #SC!SSR!BIT1!BIT2,R2    ;SET UP EXPECTED
 10146 067766  020102                        CMP     R1,R2                   ;ARE THEY EQUAL (CORRECT)
 10147 067770  001404                        BEQ     440$                    ;BR, IF CORRECT STATUS
 10151 067772                                ERRHRD  ERRNO,T38WRT,PKTSSR     ;"TSSR INCORRECT AFTER WRITE COMMAND
       067772  104456                                                                TRAP    C$ERHRD
       067774  001766                                                                .WORD   1014
       067776  071740                                                                .WORD   T38WRT
       070000  012046                                                                .WORD   PKTSSR
 10152 070002                          440$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
       070002  104406                                                                TRAP    C$CLP1
 10153 070004  013701  071034                MOV     T38BFR+6,R1             ;READ XST0 CONTENTS
 10154 070010  010102                        MOV     R1,R2                   ;SET UPR EXPECTED
 10155 070012  052702  004000                BIS     #BIT11,R2               ;SET THE WRITE LOCK ERROR BIT (XST0)
 10156 070016  020102                        CMP     R1,R2                   ;WAS THE BIT SET
 10157 070020  001404                        BEQ     450$                    ;BR, IF IT WAS (GOOD)
 10161 070022                                ERRHRD  ERRNO,T38WLE,EXPREC     ;"WRITE LOCK ERROR BIT NOT SET"
       070022  104456                                                                TRAP    C$ERHRD
       070024  001767                                                                .WORD   1015
       070026  072065                                                                .WORD   T38WLE
       070030  015474                                                                .WORD   EXPREC
 10162 070032                          450$:   CKLOOP                          ;LOOP IF SELECTED
       070032  104406                                                                TRAP    C$CLP1
 10163 070034  000137  066252                JMP     2$                      ;GO BACK TO MENU
 10164
 10165                                ;*************************************************************************
 10166                                ;       SERVO EXERCISER NO LONGER USED
 10167                                ;*************************************************************************
 10168 070040                          30$:
 10169 070040                                PRINTB  #T38MS3                 ;"EXE ANY OTHER MENU SELECTION TO STOP
       070040  012746  071645                                                   MOV     #T38MS3,-(SP)
       070044  012746  000001                                                   MOV     #1,-(SP)
       070050  010600                                                           MOV     SP,R0
       070052  104414                                                           TRAP    C$PNTB
       070054  062706  000004                                                   ADD     #4,SP
 10170 070060  004737  073342                JSR     PC,T38REST              ;SET PACKET TO INITIAL VALUES
 10171 070064  004737  015774                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 10172 070070  103405                        BCS     500$                    ;BR IF SOFT INIT = OK
 10176 070072  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 10177 070074                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
       070074  104455                                                           TRAP    C$ERDF
       070076  001770                                                           .WORD   1016
       070100  003652                                                           .WORD   SFIERR
       070102  012034                                                           .WORD   SFIMSG
 10178 070104  013737  002202  071530  500$: MOV     UNITN,T38DSW            ;SET UNIT NUMBER
 10179 070112  012704  071510                MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 10180 070116  004737  010662                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 10181 070122  103405                        BCS     510$                    ;BR, IF COMMAND ISSUED OK
 10185 070124  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 10186 070126                                ERRHRD  ERRNO WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
       070126  104456                                                           TRAP    C$ERHRD
```

TEST 10: MANUAL INTERVENTION

```
        070130  001771                                                                  .WORD   1017
        070132  005056                                                                  .WORD   WRTMSG
        070134  012034                                                                  .WORD   SFIMSG
 10187  070136                          510$:
 10188  070136  112737  000000  071021          MOVB    #0,T38BS1               ;CLEAR BIT #4
 10189  070144  112737  000020  071020          MOVB    #20,T38BSO              ;EXECUTE DRIVE TEST 22
 10190  070152  012704  071010                  MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 10191  070156  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 10192  070162  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 10193  070166  103405                          BCS     550$                    ;BR IF CARRY SET (GOOD RETURN)
 10194  070170  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 10198  070172                                  ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        070172  104455                                                                  TRAP    C$ERDF
        070174  001772                                                                  .WORD   1018
        070176  072206                                                                  .WORD   T38SSR
        070200  012046                                                                  .WORD   PKTSSR
 10199  070202                          550$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        070202  104406                                                                  TRAP    C$CLP1
 10200  070204                                  SETPRI  #PRI07                  ;RAISE THE PRIORITY
        070204  012700  000340                                                          MOV     #PRI07,R0
        070210  104441                                                                  TRAP    C$SPRI
 10201  070212  005037  070776                  CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
 10202  070216  032737  000100  177560          BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
 10203  070224  001005                          BNE     555$                    ;BRANCH IF YES
 10204  070226  005237  070776                  INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
 10205  070232  052737  000100  177560          BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
 10206  070240  012701  000060          555$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
 10207  070244  011137  071000                  MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
 10208  070250  012721  070276                  MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
 10209  070254  011137  071002                  MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
 10210  070260  012711  000340                  MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
 10211  070264                                  SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
        070264  012700  000000                                                          MOV     #PRI00,R0
        070270  104441                                                                  TRAP    C$SPRI
 10212  070272  000240          560$:   NOP                                     ;LOOP AWHILE
 10213  070274  000776                  BR      560$                            ;STAY IN "TIGHT" LOOP
 10214                                  ;+
 10215                                  ;PROCESS CONSOLE INTERRUPTS
 10216                                  ;-
 10217
 10218  070276  010046          590$:   MOV     R0,-(SP)                        ;SAVE WORK REGISTER
 10219  070300  113700  177562          MOVB    @#TTI8FR,R0             ;GET THE OPERATOR INPUT
 10220  070304  042700  000200                  BIC     #200,R0                 ;STRIP OFF PARITY BIT
 10221  070310  122700  000015                  CMPB    #15,R0                  ;IS IT A CARRIAGE RETURN ?
 10222  070314  001075                          BNE     591$                    ;JUST EXIT IF NOT
 10223  070316  012766  066252  000002          MOV     #2$,2(SP)               ;RETURN TO MASTER MENU
 10224  070324  005066  000004                  CLR     4(SP)                   ;FORCE PRIORTY 0
 10225  070330  013737  071000  000060          MOV     TVSAV2,@#TTIVEC ;RESTORE VECTOR
 10226  070336  013737  071002  000062          MOV     TPSAV2,@#TTIVEC+2       ;RESTORE SUPER PRIORTY
 10227  070344  112737  000025  071020          MOVB    #25,T38BSO              ;STOP DRIVE TEST 22
 10228  070352  112737  000000  071021          MOVB    #0,T38BS1               ;CLEAR BS1
 10229  070360  012704  071010                  MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 10230  070364  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 10231  070370  012737  176750  071004          MOV     #65000.,T38DLY          ;SET UP DELAY COUNTER
 10232  070376  004737  016250          592$:   JSR     PC,WAITF                ;DO A WAIT FOR SSR
 10233  070402  016501  000002                  MOV     TSSR(R5),R1             ;CONTENTS OF TSSR REGISTER
 10234  070406  032701  000200                  BIT     #SSR,R1                 ;CHECK FOR TSSR SET
```

TEST 10: MANUAL INTERVENTION

```
10235 070412  001017                        BNE     595$                    ;KEEP GOING IF NOT SET
10236 070414                                DELAY   250                     ;CALL DELAY ROUTINE
      070414  012727  000250                                                      MOV     #250.,(PC)+
      070420  000000                                                             .WORD   0
      070422  013727  002116                                                     MOV     L$DLY,(PC)+
      070426  000000                                                             .WORD   0
      070430  005367  177772                                                     DEC     -6(PC)
      070434  001375                                                             BNE     .-4
      070436  005367  177756                                                     DEC     -22(PC)
      070442  001367                                                             BNE     .-20
10237 070444  005337  071004                DEC     T38DLY                  ;BUMP COUNTER DOWN
10238 070450  001352                        BNE     592$                    ;BR, IF MORE TIME LEFT
10239 070452  004737  016336        595$:   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
10240 070456  103405                        BCS     580$                    ;BR IF CARRY SET (GOOD RETURN)
10241 070460  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10245 070462                                ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      070462  104455                                                             TRAP    C$ERDF
      070464  001773                                                             .WORD   1019
      070466  C72206                                                             .WORD   T38SSR
      070470  012046                                                             .WORD   PKTSSR
10246 070472                        580$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      070472  104406                                                             TRAP    C$CLP1
10247 070474  005737  070776                TST     TTION2                  ;ARE SUPER INTERRUPTS ENABLED
10248 070500  001403                        BEQ     591$                    ;BR, IF YES
10249 070502  042737  000100  177560        BIC     #100,@#TTICSR           ;RESTORE REGISTER
10250 070510  012600                591$:   MOV     (SP)+,R0                ;RESTORE REGISTER
10251 070512  000002                        RTI                             ;RETURN
10252 070514                        35$:
10253 070514  004737  073342                JSR     PC,T38REST              ;SET PACKET TO INITIAL VALUES
10254 070520  004737  015774                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
10255 070524  103405                        BCS     600$                    ;BR IF SOFT INIT = OK
10259 070526  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10260 070530                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      070530  104455                                                             TRAP    C$ERDF
      070532  001774                                                             .WORD   1020
      070534  003652                                                             .WORD   SFIERR
      070536  012034                                                             .WORD   SFIMSG
10261 070540                        600$:   CKLOOP                          ;LOOP IF SELECTED
      070540  104406                                                             TRAP    C$CLP1
10262 070542  012701  071026                MOV     @T38BFR,R1              ;ADDRESS OF MESSAGE BUFFER
10263 070546  012702  125252                MOV     @125252,R2              ;ALTERNATING 1'S AND 0'S
10264
10265 070552  010221                601$:   MOV     R2,(R1)+                ;CLEAR OUT THE MESSAGE BUFFER
10266 070554  022701  071502                CMP     @T38EB,R1               ;END OF BUFFER YET
10267 070560  001401                        BEQ     605$                    ;BR, IF AT END OF BUFFER
10268 070562  000773                        BR      601$                    ;NOT AT END KEEP GOING
10269 070564  013737  002202  071530  605$: MOV     UNITN,T38DSW            ;SET UNIT NUMBER
10270 070572  012704  071510                MOV     @T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10271 070576  004737  010662                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10272 070602  103405                        BCS     610$                    ;BR, IF COMMAND ISSUED OK
10276 070604  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10277 070606                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      070606  104456                                                             TRAP    C$ERHRD
      070610  001775                                                             .WORD   1021
      070612  005056                                                             .WORD   WRTMSG
      070614  012034                                                             .WORD   SFIMSG
10278 070616                        610$:   CKLOOP                          ;LOOP IF SELECTED
```

TEST 10: MANUAL INTERVENTION

```
        070616  104406                                                          TRAP    C$CLP1
 10279  070620  112737  000000  071021          MOVB    #0,T38BS1               ;CLEAR BIT #4
 10280  070626  112737  000024  071020          MOVB    #24,T38BS0              ;READ EXTENDED DRIVE STATUS
 10281  070634  012704  071010                  MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 10282  070640  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 10283  070644  012737  000144  071004          MOV     #100.,T38DLY            ;SET UP DELAY ROUTINE
 10284  070652  004737  016250          620$:   JSR     PC,WAITF                ;WAIT AWHILE FOR SSR TO SET
 10285  070656  016501  000002                  MOV     TSSR(R5),R1             ;SEE IF IT REALLY DID
 10286  070662  032701  000200                  BIT     #SSR,R1                 ;JUST CHECK THAT BIT
 10287  070666  001017                          BNE     630$                    ;BR, IF SSR IS SET
 10288  070670                                  DELAY   250                     ;DELAY ABOUT .25 SEC
        070670  012727  000250                                                  MOV     #250.,(PC)+
        070674  000000                                                          .WORD   0
        070676  013727  002116                                                  MOV     L$DLY,(PC)+
        070702  000000                                                          .WORD   0
        070704  005367  177772                                                  DEC     -6(PC)
        070710  001375                                                          BNE     .-4
        070712  005367  177756                                                  DEC     -22(PC)
        070716  001367                                                          BNE     .-20
 10289  070720  005337  071004                  DEC     T38DLY                  ;START DELAY COUNT DOWN
 10290  070724  001352                          BNE     620$                    ;BR, IF COUNTER IS NOT AT DONE
 10291  070726  004737  016336          630$:   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 10292  070732  103405                          BCS     650$                    ;BR IF CARRY SET (GOOD RETURN)
 10293  070734  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 10297  070736                                  ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        070736  104455                                                          TRAP    C$ERDF
        070740  001776                                                          .WORD   1022
        070742  072206                                                          .WORD   T38SSR
        070744  012046                                                          .WORD   PKTSSR
 10298  070746                  650$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
        070746  104406                                                          TRAP    C$CLP1
 10299  070750  012700  071046                  MOV     #T38BFR+20,R0           ;MESSAGE BUFFER ADDRESS
 10300  070754  005001                          CLR     R1                      ;NO HIGH ORDER ADDRESS BITS
 10301  070756  005037  003134                  CLR     KTENABLE                ;NO KT11 STUFF EITHER
 10302  070762  004737  073400                  JSR     PC,T38MBP               ;GO PRINT MESSAGE BUFFER CONTENTS
 10303  070766  000137  066252                  JMP     2$                      ;GO BACK TO MENU
 10304
 10305
 10306                          ;63$:   JMP     200                     ;TYPE ^C TO RETURN TO THE SUPERVISOR
 10307  070772                  64$:    EXIT    TST                     ;LEAVE TEST
        070772  104432                                                          TRAP    C$EXIT
        070774  003056                                                          .WORD   L10075-.
 10308
 10309                          ;+
 10310                          ;LOCAL TEXT MESSAGES FOR TEST
 10311                          ;-
 10312
 10313                          ;LOCAL STORAGE FOR THIS TEST
 10314                          ;-
 10315                          ;+
 10316                          ;LOCAL STORAGE FOR THIS TEST
 10317                          ;-
 10318
 10319  070776  000000          TTION2:         .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
 10320  071000  000000          TVSAV2: .WORD   0               ;SAVE TTI VECTOR
 10321  071002  000000          TPSAV2: .WORD   0               ;SAVE TTI PRIORITY
 10322
```

TEST 10: MANUAL INTERVENTION

```
10323 071004  000000                T38DLY: .WORD   0                       ;DELAY COUNTER FOR TEST
10325         071010                        .=<.+10>&177770
10327 071010                        T38PACKET:                              ;COMMAND PACKET FOR TEST
10328 071010  140006                        .WORD   140006                  ;WRITE SUBSYSTEM MEM. CMD.  ACK,CVC=1
10329 071012  071020                        .WORD   T38TAD                  ;ADDRESS OF CHARACTERISTICS BLOCK
10330 071014  000000                        .WORD   0
10331 071016  000012                        .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
10332 071020                        T38TAD:                         ;CHARACTERISTICS DATA BLOCK
10333 071020     000                T38BS0: .BYTE   0                       ;BSEL0 BYTE
10334 071021     000                T38BS1: .BYTE   0                       ;BSEL1 BYTE
10335 071022  000000                T38BS2: .WORD   0                       ;BSEL1 WORD
10336 071024  000000                        .WORD   0                       ;DATA
10337 071026                        T38BFR: .BLKW   150.                    ;MESSAGE BUFFER
10338 071502  000000                T38EB:  .WORD                           ;END OF BUFFER ADDRESS
10339
10340
10342         071510                        .=<.+10>&177770
10344 071510                        T38PK2:                                 ;COMMAND PACKET FOR TEST
10345 071510  140004                        .WORD   140004                  ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
10346 071512  071520                        .WORD   T38DTA                  ;ADDRESS OF SELECT DATA BLOCK
10347 071514  000000                        .WORD   0
10348 071516  000012                        .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
10349
10350
10351 071520                        T38DTA:                                 ;SELECT DATA BLOCK
10352 071520  071026                        .WORD   T38BFR                  ;ADDRESS OF MESSAGE BUFFER
10353 071522  000000                        .WORD   0
10354 071524  000400                        .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
10355 071526  000000                T38EAI: .WORD   0                       ;EAI BIT WORD
10356 071530  000000                T38DSW: .WORD   0                       ;DRIVE SELECT WORD ETC
10358         071540                        .=<.+10>&177770
10360 071540  140212                T38PK3: .WORD   140212                  ;MESSAGE BUFFER RELEASE COMMAND
10361 071542  000000                        .WORD   0                       ;NOT USED
10362 071544  000000                        .WORD   0                       ;NOT USED
10363 071546  000000                        .WORD   0                       ;NOT USED
10364 071550  000000                        .WORD   0                       ;NOT USED
10365                               ;
10366                               ;WRITE TAPE PACKET
10367                               ;
10369         071560                        .=<.+10>&177770
10371 071560  140005                T38PK4: .WORD   140005                  ;WRITE, ACK, CVC=1 COMMAND
10372 071562  000000                T38WR:  .WORD   0                       ;ADDRESS OF WRITE BUFFER
10373 071564  000000                        .WORD   0                       ;MORE ADDRESS OF WRITE BUFFER
10374 071566  000400                T38SIZ: .WORD   256.                    ;SIZE OF RECORD
10375
10376
10377
10378
10379                               ;+
10380                               ;LOCAL TEXT MESSAGES FOR TEST
10381                               ;-
10382
10383
10384
10385
10386
10387 071570     123     164     141  T38NE:  .ASCIZ  'Stand-alone Manual Intervention Not Executed'
```

TEST 10: MANUAL INTERVENTION

```
10388 071645  045  116  045  T38MS3: .ASCIZ  '%N%A Type <RETURN> To Stop Servo Exerciser, Return To Menu'
10389 071740  124  123  123  T38WRT: .ASCIZ  'TSSR Not Correct After WRITE, With WRITE PROTECT On'
10390 072024  127  122  111  T38WRL: .ASCIZ  'WRITE LOCKED Bit Not Set In XSTO'
10391 072065  127  122  111  T38WLE: .ASCIZ  'WRITE LOCK ERROR Bit Not Set In XSTO'
10392 072132  127  122  111  T38NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
10393 072206  103  157  156  T38SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
10394 072276  045  116  045  T38INT: .ASCIZ  '%N%A Interrupt Received'
10395 072326  045  116  045  T38ONL: .ASCIZ  '%N%A Drive Is Now   ON-LINE'
10396 072362  045  116  045  T38OFL: .ASCIZ  '%N%A Drive Is Now  OFF-LINE'
10397 072416  103  157  156  T38SST: .ASCIZ  'Contents Of TSSR Incorrect After MESSAGE BUFFER RELEASE'
10398 072506  045  116  045  T38MS1: .ASCIZ  '%N%AToggle ON-LINE Switch to Generate ATTENTION Interrupts'
10399 072601  045  116  045  T38MS2: .ASCIZ  '%N%AType RETURN To Return To Menu%N'
10400 072645  111  163  040  T38MSG: .ASCIZ  'Is Write-Protected Tape Mounted'
10401 072705  115  141  156  T38ID:  .ASCIZ  'Manual Intervention'
10402                         .EVEN
10403 072732 072756 073030 073056 MIMENU: .WORD   1$,2$,3$,4$,5$,6$
10404 072746 073225 073270 073337         .WORD   8$,9$,10$,0
10405
10406 072756  012  123  105  1$:      .ASCIZ  <12>'SELECT OPERATION FROM FOLLOWING OPTIONS:'
10407 073030  012  011  060  2$:      .ASCIZ  <12>'    0       Display This Menu'
10408 073056  011  061  011  3$:      .ASCIZ  '        1       Turn On All M7196 LED''s'
10409 073110  011  062  011  4$:      .ASCIZ  '        2       Turn Off All M7196 LED''s'
10410 073143  011  063  011  5$:      .ASCIZ  '        3       Offline/Online Attention'
10411 073177  011  064  011  6$:      .ASCIZ  '        4       Write Protect Test'
10412 073225  011  065  011  8$:      .ASCIZ  '        5       Print Extended Transport Status'
10413 073270  011  136  103  9$:      .ASCIZ  '        ^C      To Return to Diagnostic Supervisor'
10414 073337  000            10$:     .ASCIZ  ''
10415                         .EVEN
10416
10417                         ;+
10418                         ;LOCAL STORAGE FOR THIS TEST
10419                         ;-
10420
10421 073340 000000           T38DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
10422 073342                  T38REST:
10423 073342                          SAVREG                          ;SAVE THE REGISTERS
10424 073346 012701 071010            MOV     #T38PACKET,R1           ;START OF THE PACKET
10425 073352 012721 140206            MOV     #140206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
10426 073356 012721 071020            MOV     #T38TAD,(R1)+           ;ADDRESS OF DATA BLOCK
10427 073362 005021                   CLR     (R1)+                   ;EXTENDED ADDRESS
10428 073364 012721 000006            MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
10429 073370 005021                   CLR     (R1)+                   ;CLEAR BSEL0 AND BSEL1
10430 073372 005021                   CLR     (R1)+                   ;CLEAR SEL2
10431 073374 005011                   CLR     (R1)                    ;CLEAR DATA AREA
10432 073376 000207                   RTS     PC                      ;RETURN
10433
10434
10435                         ;+
10436                         ;
10437                         ;THIS ROUTINE PRINTS THE CONTENTS OF
10438                         ;THE 256 BYTE MESSAGE BUFFER RETURNED BY THE
10439                         ;TSV-05.
10440                         ;
10441                         ;INPUT:
10442                         ;
10443                         ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
10444                         ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
```

TEST 10: MANUAL INTERVENTION

```
10445                                   ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
10446                                   ;
10447                                   ;
10448                                   ;-
10449
10450 073400                            T38MBP:
10451 073400                                    SAVREG                          ;SAVE THE REGISTERS
10452 073404  010005                              MOV     R0,R5                 ;SAVE LOW ORDER ADDRESS
10453 073406  005737  003134                      TST     KTENABLE              ;ADDRESS ABOVE 28K?
10454 073412  001001                              BNE     910$                  ;BR IF YES
10455 073414  005001                              CLR     R1                    ;SET HIGH ORDER ADDRESS TO 0
10456 073416  010103                    910$:     MOV     R1,R3                 ;SAVE HIGH ORDER ADDRESS
10457 073420  006100                              ROL     R0                    ;SHIFT BIT15 TO C BIT
10458 073422  006101                              ROL     R1                    ;SHIFT TO HIGH ORDER FOR PRINTOUT
10459 073424                                      PRINTX  #T38AS0,R1,R5         ;PRINT MESSAGE BUFFER ADDRESS
      073424  010546                                                                  MOV     R5,-(SP)
      073426  010146                                                                  MOV     R1,-(SP)
      073430  012746  073702                                                          MOV     #T38AS0,-(SP)
      073434  012746  000003                                                          MOV     #3,-(SP)
      073440  010600                                                                  MOV     SP,R0
      073442  104415                                                                  TRAP    C$PNTX
      073444  062706  000010                                                          ADD     #10,SP
10460 073450                                      PRINTX  #T38AS1               ;PRINT HEADER FOR CONTENTS
      073450  012746  073747                                                          MOV     #T38AS1,-(SP)
      073454  012746  000001                                                          MOV     #1,-(SP)
      073460  010600                                                                  MOV     SP,R0
      073462  104415                                                                  TRAP    C$PNTX
      073464  062706  000004                                                          ADD     #4,SP
10461 073470  010501                              MOV     R5,R1                 ;COPY LOW ORDER ADDRESS
10462 073472  010300                              MOV     R3,R0                 ;COPY HIGH ORDER ADDRESS
10463 073474  001403                              BEQ     913$                  ;BR IF NOT ABOVE 28K
10464 073476  004737  017316                      JSR     PC,SETMAP             ;SETUP PAR ADDRESS IN R0
10465 073502  010005                              MOV     R0,R5                 ;GET PAR FORMAT ADDRESS ABOVE 28K
10466 073504  010537  074050            913$:     MOV     R5,T38CNT             ;HOLD ADDRESS
10467 073510  011504                    911$:     MOV     (R5),R4               ;GET BUFFER ENTRY
10468 073512  022704  125252                      CMP     #125252,R4            ;CHECK FOR NO LOAD CONDITION
10469 073516  001417                              BEQ     912$                  ;BR, IF BUFFER WASN'T LOADED
10470 073520  010403                              MOV     R4,R3                 ;MAKE COPY
10471 073522  042704  170377                      BIC     #170377,R4            ;ONLY BITS 11,10,9 AND 8 ARE SAVED
10472 073526  000241                              CLC                           ;CLEAR CARRY
10473 073530  006004                              ROR     R4                    ;11 TO 10 BIT POSITION
10474 073532  006004                              ROR     R4                    ;10 TO 9 BIT POSITION
10475 073534  006004                              ROR     R4                    ;9 TO 8 BIT POSITON
10476 073536  006004                              ROR     R4                    ;8 TO 7 BIT POSITION
10477 073540  042703  177760                      BIC     #177760,R3            ;ONLY BITS 3,2,1 AND 0 ARE SAVED
10478 073544  060403                              ADD     R4,R3                 ;"OR'EM TOGETHER
10479 073546  010325                              MOV     R3,(R5)+              ;PUT BACK IN BUFFER
10480 073550  020527  0/1502                      CMP     R5,#T38EB             ;END OF BUFFER YET
10481 073554  001355                              BNE     911$                  ;BR, IF NOT AT END YET
10482 073556  013705  074050            912$:     MOV     T38CNT,R5             ;PUT ADDRESS BACK
10483 073562  012704  000001                      MOV     #1,R4                 ;START BYTE NUMBER AT ONE
10484 073566                            915$:     PRINTX  #T38ASN,R4,(R5)+              ;PRT MEM BUFFER W/NEWLINE
      073566  012546                                                                  MOV     (R5)+,-(SP)
      073570  010446                                                                  MOV     R4,-(SP)
      073572  012746  074024                                                          MOV     #T38ASN,-(SP)
      073576  012746  000003                                                          MOV     #3,-(SP)
      073602  010600                                                                  MOV     SP,R0
```

TEST 10: MANUAL INTERVENTION

```
        073604  104415                                                                  TRAP    C$PNTX
        073606  062706  000010                                                          ADD     #10,SP
 10485  073612  005037  074050                  CLR     T38CNT                  ;CLEAR COUNTER
 10486  073616  000412                          BR      921$                    ;SKIP OTHER PRINT
 10487  073620                          920$:   PRINTX  #T38ASC,R4,(R5)+        ;PRINT THE CONTENTS OF MEMORY BUFFER
        073620  012546                                                          MOV     (R5)+,-(SP)
        073622  010446                                                          MOV     R4,-(SP)
        073624  012746  074005                                                  MOV     #T38ASC,-(SP)
        073630  012746  000003                                                  MOV     #3,-(SP)
        073634  010600                                                          MOV     SP,R0
        073636  104415                                                          TRAP    C$PNTX
        073640  062706  000010                                                  ADD     #10,SP
 10488  073644  005237  074050          921$:   INC     T38CNT                  ;BUMP COUNTER
 10489  073650  005204                          INC     R4              ;NUMBER OF THE NEXT
 10490  073652  020427  000200                  CMP     R4,#128.                ;DONE ALL YET ?
 10491  073656  003010                          BGT     50$             ;BRANCH IF ALL DONE
 10492  073660  023727  074050  000004          CMP     T38CNT,#4               ;DONE FOUR YET
 10493  073666  001401                          BEQ     925$                    ;BR, IF THREE DONE
 10494  073670  C00753                          BR      920$            ;KEEP GOING
 10495  073672  005037  074050          925$:   CLR     T38CNT                  ;CLEAR COUNTER
 10496  073676  000733                          BR      915$                    ;PRINT WITH NEW LINE
 10497  073700  000207                  50$:    RTS     PC              ;RETURN
 10498
 10499  073702     045     116     045  T38AS0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
 10500  073747     045     116     045  T38AS1: .ASCIZ  '%N%A Message Buffer Contents.'
 10501  074005     045     101     040  T38ASC: .ASCIZ  '%A  %D4%A: %03'
 10502  074024     045     116     045  T38ASN: .ASCIZ  '%N%A Byte%D4%A: %03'
 10503                                   .EVEN
 10504  074050  000000                  T38CNT: .WORD                   ;COUNTER FOR PRINT
 10505  074052                          ENDTST
        074052                                                                  L10075:
        074052  104401                                                                  TRAP    C$ETST
 10506                          .SBTTL  TEST 11: CONFIGURATION TYPEOUT
 10507
 10508                  ;THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL
 10509                  ;THE CONFIGURATION OF THE M7196 MODULE AND TSV05 SUBSYSTEM. SPECIFICALLY,
 10510                  ;THE FOLLOWING INFORMATION IS PRESENTED:
 10511                  ;
 10512                  ;
 10513                  ; 1.0 STATE OF THE EXTENDED FEATURES SWITCH ON THE M7196: ON (EXTENDED
 10514                  ;     FEATURES ENABLED) OR OFF (EXTENDED FEATURES DISABLED).
 10515                  ;
 10516                  ; 2.0 STATE OF THE BUFFERING ENABLE SWITCH ON THE M7196: ON
 10517                  ;     (BUFFERING ENABLED) OR OFF (BUFFERING DISABLED).
 10518                  ;
 10519                  ; 3.0 MICROCODE REVSION LEVEL OF THE M7196.
 10520                  ;
 10521                  ; 4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.
 10522                  ;
 10523                  ; 5.0 UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED)
 10524                  ;     OF EACH CONNECTED TRANSPORT. IN ADDITION, THE PROGRAM WILL INDICATE,
 10525                  ;     FOR EACH ON-LINE TRANSPORT, WHETHER OR NOT IT IS EQUIPPED WITH THE
 10526                  ;     EXTENDED TAPE STATUS READOUT FEATURE.
 10527                  ;
 10528                  ;
 10529                  ;THE OPERATOR IS EXPECTED TO READ THE PRINTOUT AND VERIFY THAT IT MATCHES
 10530                  ;THE ACTUAL CONFIGURATION AT HAND.  IF, FOR EXAMPLE, THE PROGRAM INDICATES
```

TEST 11: CONFIGURATION TYPEOUT

```
10531                                   ;THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN FACT ONLY ONE IS PRESENT.
10532                                   ;THE OPERATOR MUST INTERPRET THIS AS AN ERROR AND ATTEMPT TO FIND THE
10533                                   ;CAUSE (BAD CABLE, FAULTY UNIT-SELECT DECODING IN THE TRANSPORT, ETC.).
10534                                   ;[SINCE THE CONTROLLER CAN ONLY ACCESS UNIT 0 IF IT IS IN "STANDARD"
10535                                   ;MODE, THE PROGRAM WILL FORCE THE MODULE INTO EXTENDED MODE VIA THE
10536                                   ;WRITE SUBSYSTEM MEMORY COMMAND IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]
10537                                   ;
10538                                   ;
10539                                   ;THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO THE
10540                                   ;DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE
10541                                   ;SELECTED (E.G., FROM THE INITIAL STARTUP DIALOG), THE ROUTINE WILL BE
10542                                   ;REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.
10543                                   ;
10544 074054                                    BGNTST
      074054                                                                    T11::
10549 074054                            632$:   RFLAGS  R0                      ;GET OPERATOR FLAGS
      074054  104421                                                                   TRAP    C$RFLA
10550 074056  001403                            BEQ     10$                     ;BR, IF OK TO RUN
10551 074060  C12700  076333                    MOV     #T39NE,R0               ;"TEST NOT EXECUTED"
10552 074064  000402                            BR      11$                     ;JUMP OUT OF TEST IF NOT
10553 074066  012700  077534              10$:  MOV     #TST39ID,R0             ;TEST ID MESSAGE
10554 074072  004737  016510              11$:  JSR     PC,TSTSETUP             ;DO THE COMMON SETUP
10555 074076  004737  017762                    JSR     PC,CHKMAN               ;IS MANUAL INTERVENTION ALLOWED?
10556 074102  103402                            BCS     20$                     ;BR, IF MANUAL INTERVENTION ALLOWED
10557 074104  000137  075336                    JMP     64$                     ;JUMP TO OUT IF NOT
10558 074110                            20$:
10559 074110  004737  015774                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
10560 074114  103405                            BCS     25$                     ;BR IF SOFT INIT = OK
10564 074116  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10565 074120                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      074120  104455                                                                    TRAP    C$ERDF
      074122  002115                                                                   .WORD   1101
      074124  003652                                                                   .WORD   SFIERR
      074126  012034                                                                   .WORD   SFIMSG
10566 074130                            25$:    CKLOOP                                  ;LOOP IF SELECTED
      074130  104406                                                                   TRAP    C$CLP1
10567 074132  013737  002202  076300            MOV     UNITN,T39DSW            ;SET UNIT NUMBER
10568 074140  012704  076260                    MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10569 074144  004737  010662                    JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10570 074150  103405                            BCS     50$                     ;BR, IF COMMAND ISSUED OK
10574 074152  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10575 074154                                    ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      074154  104456                                                                   TRAP    C$ERHRD
      074156  002116                                                                   .WORD   1102
      074160  005056                                                                   .WORD   WRTMSG
      074162  012034                                                                   .WORD   SFIMSG
10576 074164                            50$:    CKLOOP                          ;LOOP IF SELECTED
      074164  104406                                                                   TRAP    C$CLP1
10577 074166  013701  075610                    MOV     T39BFR+12,R1            ;GET XST2 STATUS FROM MESSAGE BUFFER
10578 074172                                    PRINTX  #T39SFS                 ;"STATE OF EXTENDED FEATURES SW ="
      074172  012746  077252                                                            MOV     #T39SFS,-(SP)
      074176  012746  000001                                                            MOV     #1,-(SP)
      074202  010600                                                                    MOV     SP,R0
      074204  104415                                                                    TRAP    C$PNTX
      074206  062706  000004                                                            ADD     #4,SP
10579 074212  032701  000200                    BIT     #BIT7,R1                ;CHECK STATE OF E.F.S.
10580 074216  001011                            BNE     100$                    ;BR, IF EXT. FEA. SW. IS ON
```

TEST 11: CONFIGURATION TYPEOUT

```
10581 074220                          PRINTX  #T390FF                 ;" OFF"
      074220  012746  077376                                                  MOV     #T390FF,-(SP)
      074224  012746  000001                                                  MOV     #1,-(SP)
      074230  010600                                                          MOV     SP,R0
      074232  104415                                                          TRAP    C$PNTX
      074234  062706  000004                                                  ADD     #4,SP
10582 074240  000410                          BR      110$                    ;SKIP OTHER PRINT STATEMENT
10583 074242                        100$:     PRINTX  #T390N                  ;" ON "
      074242  012746  077405                                                  MOV     #T390N,-(SP)
      074246  012746  000001                                                  MOV     #1,-(SP)
      074252  010600                                                          MOV     SP,R0
      074254  104415                                                          TRAP    C$PNTX
      074256  062706  000004                                                  ADD     #4,SP
10584 074262                        110$:     PRINTX  #T39SBS                 ;"STATE OF BUFFERING SWITCH ="
      074262  012746  077324                                                  MOV     #T39SBS,-(SP)
      074266  012746  000001                                                  MOV     #1,-(SP)
      074272  010600                                                          MOV     SP,R0
      074274  104415                                                          TRAP    C$PNTX
      074276  062706  000004                                                  ADD     #4,SP
10585 074302  032701  000100                BIT     #BIT6,R1                ;CHECK STATE OF BUFFERING SW
10586 074306  001011                        BNE     120$                    ;BR, IF BUFFERING IS ON
10587 074310                                PRINTX  #T390FF                 ;" OFF"
      074310  012746  077376                                                  MOV     #T390FF,-(SP)
      074314  012746  000001                                                  MOV     #1,-(SP)
      074320  010600                                                          MOV     SP,R0
      074322  104415                                                          TRAP    C$PNTX
      074324  062706  000004                                                  ADD     #4,SP
10588 074330  000410                        BR      130$                    ;SKIP OTHER PRINT STATEMENT
10589 074332                        120$:   PRINTX  #T390N                  ;" ON "
      074332  012746  077405                                                  MOV     #T390N,-(SP)
      074336  012746  000001                                                  MOV     #1,-(SP)
      074342  010600                                                          MOV     SP,R0
      074344  104415                                                          TRAP    C$PNTX
      074346  062706  000004                                                  ADD     #4,SP
10590 074352  042701  177700          130$:   BIC     #177700,R1              ;ONLY LEAVE MICROCODE REV LEVEL
10591 074356  010137  077472                  MOV     R1,T39RL                ;LOAD UP REV LEVEL
10592 074362                                PRINTX  #T39MCL,T39RL           ;"MICROCODE REVISION LEVEL =000XXX"
      074362  013746  077472                                                  MOV     T39RL,-(SP)
      074366  012746  077414                                                  MOV     #T39MCL,-(SP)
      074372  012746  000002                                                  MOV     #2,-(SP)
      074376  010600                                                          MOV     SP,R0
      074400  104415                                                          TRAP    C$PNTX
      074402  062706  000006                                                  ADD     #6,SP
10593 074406  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
10594 074412  103405                        BCS     140$                    ;BR IF SOFT INIT = OK
10598 074414  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10599 074416                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      074416  104455                                                          TRAP    C$ERDF
      074420  002117                                                          .WORD   1103
      074422  003652                                                          .WORD   SFIERR
      074424  012034                                                          .WORD   SFIMSG
10600 074426                        140$:   CKLOOP                                  ;LOOP IF SELECTED
      074426  104406                                                          TRAP    C$CLP1
10601 074430  013737  002202  076300        MOV     UNITN,T39DSW            ;SET UNIT NUMBER
10602 074436  012704  076260                MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10603 074442  004737  010662                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10604 074446  103405                        BCS     150$                    ;BR, IF COMMAND ISSUED OK
```

TEST 11: CONFIGURATION TYPEOUT

```
10608 074450  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10609 074452                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      074452  104456                                                        TRAP    C$ERHRD
      074454  002120                                                        .WORD   1104
      074456  005056                                                        .WORD   WRTMSG
      074460  012034                                                        .WORD   SFIMSG
10610 074462                        150$:   CKLOOP                          ;LOOP IF SELECTED
      074462  104406                                                        TRAP    C$CLP1
10611 074464  005737  002226                  TST     EXTFEA                  ;CHECK FOR EXTENDED FEATURES SW SWITCH
10612 074470  001036                          BNE     174$                    ;BR IF SWITCH IS ON
10613 074472  112737  000200  075571          MOVB    #200,T39BS1             ;WRITE MISCELLANEOUS CONT/READ STATUS
10614 074500  112737  000010  075570          MOVB    #10,T39BSO              ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
10615 074506  012704  075560                  MOV     #T39PACKET,R4           ;WRITE SUBSYS MEM PACKET
10616 074512  010465  000000                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
10617 074516  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
10618 074522  103405                          BCS     160$                    ;BR, IF NO ERROR
10619 074524  010001                          MOV     R0,R1                   ;ERROR, SAVE TSSR
10623 074526                                ERRHRD  ERRNO,T39NBA,PKTSSR     ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      074526  104456                                                        TRAP    C$ERHRD
      074530  002121                                                        .WORD   1105
      074532  077035                                                        .WORD   T39NBA
      074534  012046                                                        .WORD   PKTSSR
10624 074536                        160$:   CKLOOP                          ;LOOP IF SELECTED
      074536  104406                                                        TRAP    C$CLP1
10625 074540  012704  076260                  MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10626                               ;*************************************************************************
10627                               ;
10628                               ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
10629                               ;
10630                               ;*************************************************************************
10631
10632 074544  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10633 074550  103405                          BCS     170$                    ;BR, IF COMMAND ISSUED OK
10637 074552  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10638 074554                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      074554  104456                                                        TRAP    C$ERHRD
      074556  002122                                                        .WORD   1106
      074560  005056                                                        .WORD   WRTMSG
      074562  012034                                                        .WORD   SFIMSG
10639 074564                        170$:   CKLOOP                          ;SCOPE LOOP
      074564  104406                                                        TRAP    C$CLP1
10640 074566  005037  002202          174$:   CLR     UNITN                   ;SET TO DRIVE 0
10641 074572  013737  002202  076300  175$:   MOV     UNITN,T39DSW            ;SET UNIT NUMBER
10642 074600  012704  076260                  MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10643 074604  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10644 074610  103405                          BCS     180$                    ;BR, IF COMMAND ISSUED OK
10648 074612  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10649 074614                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      074614  104456                                                        TRAP    C$ERHRD
      074616  002123                                                        .WORD   1107
      074620  005056                                                        .WORD   WRTMSG
      074622  012034                                                        .WORD   SFIMSG
10650 074624                        180$:   CKLOOP                          ;LOOP IF SELECTED
      074624  104406                                                        TRAP    C$CLP1
10651
10652 074626  016501  000002          190$:   MOV     TSSR(R5),R1             ;GET TSSR STATUS
10653 074632  032701  000100                  BIT     #OFL,R1                 ;CHECK FOR OFF-LINE
```

TEST 11: CONFIGURATION TYPEOUT

```
10654 074636  001414                        BEQ     200$                ;BR, IF DRIVE IS ON-LINE
10655 074640                                PRINTX  #T390F2,UNITN       ;"DRIVE NUMBER XX IS OFF-LINE"
      074640  013746  002202                                            MOV     UNITN,-(SP)
      074644  012746  076574                                            MOV     #T390F2,-(SP)
      074650  012746  000002                                            MOV     #2,-(SP)
      074654  010600                                                    MOV     SP,R0
      074656  104415                                                    TRAP    C$PNTX
      074660  062706  000006                                            ADD     #6,SP
10656 074664  000137  075220                        JMP     250$                ;DO NOT TRY TO GET ANYMORE INFO.
10657 074670                          200$:         PRINTX  #T390N2,UNITN       ;"DRIVE NUMBER XX IS ON-LINE"
      074670  013746  002202                                            MOV     UNITN,-(SP)
      074674  012746  076640                                            MOV     #T390N2,-(SP)
      074700  012746  000002                                            MOV     #2,-(SP)
      074704  010600                                                    MOV     SP,R0
      074706  104415                                                    TRAP    C$PNTX
      074710  062706  000006                                            ADD     #6,SP
10658 074714  013701  075604                        MOV     T39BFR+6,R1         ;READ EXTENDED STATUS (XSTO)
10659 074720  032701  000004                        BIT     #BIT2,R1            ;IS DRIVE WTITE PROTECTED
10660 074724  C01013                        BNE     210$                ;BR, IF WRITE PROTECTED
10661 074726                                PRINTX  #T39WPN,UNITN       ;"DRIVE NUMBER IS NOT WRT PRO"
      074726  013746  002202                                            MOV     UNITN,-(SP)
      074732  012746  076756                                            MOV     #T39WPN,-(SP)
      074736  012746  000002                                            MOV     #2,-(SP)
      074742  010600                                                    MOV     SP,R0
      074744  104415                                                    TRAP    C$PNTX
      074746  062706  000006                                            ADD     #6,SP
10662 074752  000412                        BR      220$                ;SKIP OVER
10663 074754                          210$:         PRINTX  #T39WRT,UNITN       ;"DRIVE NUMBER XX IS WRT PRO"
      074754  013746  002202                                            MOV     UNITN,-(SP)
      074760  012746  076703                                            MOV     #T39WRT,-(SP)
      074764  012746  000002                                            MOV     #2,-(SP)
      074770  010600                                                    MOV     SP,R0
      074772  104415                                                    TRAP    C$PNTX
      074774  062706  000006                                            ADD     #6,SP
10664 075000  012737  125252  075676  220$:  MOV     #125252,T39BFR+100  ;SET 1 LOC TO KNOWN VALUE
10665 075006  112737  000000  075571        MOVB    #0,T39BS1           ;EXTENDED TAPE STATUS
10666 075014  112737  000024  075570        MOVB    #24,T39BS0          ;EXTENDED TAPE STATUS
10667 075022  012704  075560                MOV     #T39PACKET,R4       ;WRITE SUBSYS MEM PACKET
10668 075026  010465  000000                MOV     R4,TSDB(R5)         ;ISSUE COMMAND
10669 075032  012737  000144  075552        MOV     #100.,T39DLY        ;SET UP DELAY ROUTINE
10670 075040  004737  016250          222$: JSR     PC,WAITF            ;WAIT AWHILE FOR SSR TO SET
10671 075044  016501  000002                MOV     TSSR(R5),R1         ;SEE IF IT REALLY DID
10672 075050  032701  000200                BIT     #SSR,R1             ;JUST CHECK THAT BIT
10673 075054  001017                        BNE     225$                ;BR, IF SSR IS SET
10674 075056                                DELAY   250                 ;DELAY ABOUT .25 SEC
      075056  012727  000250                                            MOV     #250.,(PC)+
      075062  000000                                                    .WORD   0
      075064  013727  002116                                            MOV     L$DLY,(PC)+
      075070  000000                                                    .WORD   0
      075072  005367  177772                                            DEC     -6(PC)
      075076  001375                                                    BNE     .-4
      075100  005367  177756                                            DEC     -22(PC)
      075104  001367                                                    BNE     .-20
10675 075106  005337  075552                        DEC     T39DLY              ;START DELAY COUNT DOWN
10676 075112  001352                        BNE     222$                ;BR, IF COUNTER IS NOT AT DONE
10677 075114  004737  016336          225$: JSR     PC,CHKTSSR          ;WAIT FOR SSR
10678 075120  103405                        BCS     230$                ;BR, IF NO ERROR
```

TEST 11: CONFIGURATION TYPEOUT

```
10679 075122  010001                          MOV     R0,R1                   ;ERROR, SAVE TSSR
10683 075124                                  ERRHRD  ERRNO,T39NBA,PKTSSR     ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      075124  104456                                                                  TRAP    C$ERHRD
      075126  002124                                                                  .WORD   1108
      075130  077035                                                                  .WORD   T39NBA
      075132  012046                                                                  .WORD   PKTSSR
10684 075134                          230$:   CKLOOP                          ;LOOP IF SELECTED
      075134  104406                                                                  TRAP    C$CLP1
10685 075136  023727  075676  125252          CMP     T39BFR+100,#125252      ;DID LOC GET OVER WRITTEN
10686 075144  001013                          BNE     240$                    ;BR, IF IT DIDN'T GET ETC.
10687 075146                                  PRINTX  #T39ETN,UNITN           ;"DRIVE DOESN'T HAVE EXT TAPE STATUS
      075146  013746  002202                                                          MOV     UNITN,-(SP)
      075152  012746  076501                                                          MOV     #T39ETN,-(SP)
      075156  012746  000002                                                          MOV     #2.-(SP)
      075162  010600                                                                  MOV     SP,R0
      075164  104415                                                                  TRAP    C$PNTX
      075166  062706  000006                                                          ADD     #6.SP
10688 075172  000412                          BR      250$                    ;SKIP OVER
10689 075174                          240$:   PRINTX  #T39ETS,UNITN           ;"DRIVE HAS EXT TAPE STATUS"
      075174  013746  002202                                                          MOV     UNITN,-(SP)
      075200  012746  076412                                                          MOV     #T39ETS,-(SP)
      075204  012746  000002                                                          MOV     #2.-(SP)
      075210  010600                                                                  MOV     SP,R0
      075212  104415                                                                  TRAP    C$PNTX
      075214  062706  000006                                                          ADD     #6.SP
10690 075220  005237  002202          250$:   INC     UNITN                   ;BUMP DRIVE NUMBER
10691 075224  023727  002202  000003          CMP     UNITN,#3                ;AT END OF DRIVES YET
10692 075232  001402                          BEQ     631$                    ;BR, IF NO MORE DRIVES
10693 075234  000137  074572                  JMP     175$                    ;DO NEXT DRIVE
10694 075240  012700  075342          631$:   MOV     #NUMENU,R0              ;MENU OF MANUAL INTERVENTIONS
10695 075244  012701  000001                  MOV     #1,R1                   ;MAXIMUM ALLOWED SELECTION
10696 075250  004737  017540                  JSR     PC,GETSEL               ;GO GET THE OPERATORS SELECTION
10697 075254  010004                          MOV     R0,R4                   ;GET NUMBER FROM ROUTINE
10698 075256  006304                          ASL     R4                      ;CONVERT TO WORD OFFSET
10699 075260  000174  075264                  JMP     @6$(R4)                 ;JUMP TO PROPER LOOP
10700 075264  075240                  6$:     .WORD   631$                    ;RETYPE THE MENU
10701 075266  074054                          .WORD   632$                    ;RERUN THE TEST
10702 075270                          63$:    SETPRI  #PRI00
      075270  012700  000000                                                                  MOV     #PRI00,R0
      075274  104441                                                                          TRAP    C$SPRI
10703 075276                                  printf  #T38MS2
      075276  012746  072601                                                                  MOV     #T38MS2,-(SP)
      075302  012746  000001                                                                  MOV     #1,-(SP)
      075306  010600                                                                          MOV     SP,R0
      075310  104417                                                                          TRAP    C$PNTF
      075312  062706  000004                                                                  ADD     #4.SP
10704 075316                                  PRINTX  #T39NFL                 ;NEW LINE
      075316  012746  016330                                                                  MOV     #T39NFL,-(SP)
      075322  012746  000001                                                                  MOV     #1,-(SP)
      075326  010600                                                                          MOV     SP,R0
      075330  104415                                                                          TRAP    C$PNTX
      075332  062706  000004                                                                  ADD     #4.SP
10705 075336                          64$:    EXIT    TST                     ;EXIT THIS SECTION
      075336  104432                                                                          TRAP    C$EXIT
      075340  002222                                                                          .WORD   L10076-.
10706                                 ;+
10707                                 ;LOCAL TEXT MESSAGES FOR TEST
```

TEST 11: CONFIGURATION TYPEOUT

```
10708                                   ;-
10709
10710                                   ;LOCAL STORAGE FOR THIS TEST
10711                                   ;-
10712 075342  075356  075430  075456  NUMENU: .WORD   1$,2$,3$,4$,5$,0
10713 075356     012     123     105  1$:     .ASCIZ  <12>'SELECT OPERATION FROM FOLLOWING OPTIONS:'
10714 075430     012     011     060  2$:     .ASCIZ  <12>'   0       Display This Menu'
10715 075456     011     061     011  3$:     .ASCIZ  '       1       Rerun this test'
10716 075501     011     136     103  4$:     .ASCIZ  '       ^C      To Return to Diagnostic Supervisor'
10717 075550     000                  5$:     .ASCIZ  ''
10718                                           .EVEN
10719
10720 075552  000000                  T39DLY: .WORD   0                               ;DELAY COUNTER FOR TEST
10722         075560                          .=<.+10>&177770
10724 075560                          T39PACKET:                              ;COMMAND PACKET FOR TEST
10725 075560  140006                          .WORD   140006                  ;WRITE SUBSYSTEM MEM. CMD.  AC    VC=1
10726 075562  075570                          .WORD   T39TAD                  ;ADDRESS OF CHARACTERISTICS BLOCK
10727 075564  000000                          .WORD   0
10728 075566  C00012                          .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
10729 075570                          T39TAD:                         ;CHARACTERISTICS DATA BLOCK
10730 075570     000                  T39BS0: .BYTE   0                       ;BSEL0 BYTE
10731 075571     000                  T39BS1: .BYTE   0                       ;BSEL1 BYTE
10732 075572  000000                  T39BS2: .WORD   0                       ;BSEL1 WORD
10733 075574  000000                          .WORD   0                       ;DATA
10734 075576                          T39BFR: .BLKW   150.                    ;MESSAGE BUFFER
10735
10736
10738         076260                          .=<.+10>&177770
10740 076260                          T39PK2:                                 ;COMMAND PACKET FOR TEST
10741 076260  140004                          .WORD   140004                  ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
10742 076262  076270                          .WORD   T39DTA                  ;ADDRESS OF SELECT DATA BLOCK
10743 076264  000000                          .WORD   0
10744 076266  000012                          .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
10745
10746
10747 076270                          T39DTA:                                 ;SELECT DATA BLOCK
10748 076270  075576                          .WORD   T39BFR                  ;ADDRESS OF MESSAGE BUFFER
10749 076272  000000                          .WORD   0
10750 076274  000400                          .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
10751 076276  000000                  T39EAI: .WORD   0                       ;EAI BIT WORD
10752 076300  000000                  T39DSW: .WORD   0                       ;DRIVE SELECT WORD ETC
10754         076310                          .=<.+10>&177770
10756 076310  140012                  T39PK3: .WORD   140012                  ;MESSAGE BUFFER RELEASE COMMAND
10757 076312  000000                          .WORD   0                       ;NOT USED
10758                                   ;
10759                                   ;WRITE TAPE PACKET
10760                                   ;
10762         076320                          .=<.+10>&177770
10764 076320  140005                  T39PK4: .WORD   140005                  ;WRITE, ACK, CVC=1 COMMAND
10765 076322  000000                  T39WR:  .WORD   0                       ;ADDRESS OF WRITE BUFFER
10766 076324  000000                          .WORD   0                       ;MORE ADDRESS OF WRITE BUFFER
10767 076326  000400                  T39SIZ: .WORD   256.                    ;SIZE OF RECORD
10768
10769
10770
10771
10772                                   ;+
```

TEST 11: CONFIGURATION TYPEOUT

```
10773                                   ;LOCAL TEXT MESSAGES FOR TEST
10774                                   ;-
10775
10776
10777
10778
10779 076330     045     116     000    T39NFL: .ASCIZ  '%N'
10780 076333     123     164     141    T39NE:  .ASCIZ  'Stand-alone Configuration Typeout Not Executed'
10781 076412     045     116     045    T39ETS: .ASCIZ  '%N%A Extended Tape Status Availiable, Drive Number %D2'
10782 076501     045     116     045    T39ETN: .ASCIZ  '%N%A Extended Tape Status NOT Availiable, Drive Number %D2'
10783 076574     045     116     045    T390F2: .ASCIZ  '%N%A Drive Number %D2%A Is Off-Line'
10784 076640     045     116     045    T390N2: .ASCIZ  '%N%A Drive Number %D2%A Is On-Line'
10785 076703     045     116     045    T39WRT: .ASCIZ  '%N%A Drive Number %D2%A Is Write Protected'
10786 076756     045     116     045    T39WPN: .ASCIZ  '%N%A Drive Number %D2%A Is NOT Write Protected'
10787 077035     127     122     111    T39NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
10788 077111     103     157     156    T39SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
10789 077201     045     116     045    T390UT: .ASCIZ  '%N%A Type ^C to Return to the Supervisor'
10790 077252     045     116     045    T39SFS: .ASCIZ  '%N%A State Of Extended Features Switch  ='
10791 077324     045     116     045    T39SBS: .ASCIZ  '%N%A State Of Buffering Switch          ='
10792 077376     045     101     040    T390FF: .ASCIZ  '%A OFF'
10793 077405     045     101     040    T390N:  .ASCIZ  '%A ON '
10794 077414     045     116     045    T39MCL: .ASCIZ  '%N%A M7196 Microcode Revision Level     =%02'
10795                                           .EVEN
10796 077472  000000                    T39RL:  .WORD   0
10797                                           .EVEN
10798                                           .EVEN
10799
10800                                   ;+
10801                                   ;LOCAL STORAGE FOR THIS TEST
10802                                   ;-
10803
10804 077474  000000                    T39DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
10805 077476                            T39REST:
10806 077476                                    SAVREG                  ;SAVE THE REGISTERS
10807 077502  012701  075560                     MOV     @T39PACKET,R1   ;START OF THE PACKET
10808 077506  012721  140006                     MOV     #140006,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
10809 077512  012721  075570                     MOV     @T39TAD,(R1)+   ;ADDRESS OF DATA BLOCK
10810 077516  005021                             CLR     (R1)+           ;EXTENDED ADDRESS
10811 077520  012721  000006                     MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
10812 077524  005021                             CLR     (R1)+           ;CLEAR BSEL0 AND BSEL1
10813 077526  005021                             CLR     (R1)+           ;CLEAR SEL2
10814 077530  005011                             CLR     (R1)            ;CLEAR DATA AREA
10815 077532  000207                             RTS     PC              ;RETURN
10816
10817                                   ;+
10818                                   ;LOCAL TEXT MESSAGES FOR TEST
10819                                   ;-
10820
10821 077534     103     157     156    TST39ID:        .ASCIZ  'Configuration Typeout'
10822                                           .EVEN
10823 077562                                    ENDTST
      077562                                                            L10076:
      077562  104401                                                              TRAP    C$ETST
10824
10825                                           .SBTTL  TEST 12: SCOPE LOOPS
10826
10827
```

TEST 12: SCOPE LOOPS

```
                                ;*
                                ;
                                ;
                                ;       THIS IS A STANDALONE ROUTINE PROVIDING A NUMBER OF TIGHT "SCOPE
                                ;       LOOPS" USEFUL FOR DEBUGGING BASIC REGISTER ACCESS PROBLEMS WITH
                                ;       THE M7196 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. 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     TSDBX (TSSR HIGH BYTE) WRITE ACCESS
                                ;                         (EXTENDED FEATURES SWITCH MUST BE ON
                                ;                          TO USE SELECTION CODE 7)
                                ;                 8     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 MENU LEVEL.
                                ;
                                ;-
                                ;
                                ;
077564                                  BGNTST
077564                                                                  T12::
077564                                  RFLAGS  R0                      ;GET OPERATOR FLAGS
077564  104421                                                                  TRAP    C$RFLA
077566  001403                          BEQ     1$                      ;BR, IF OK TO RUN
077570  012700  101155                  MOV     @T40NE,R0               ;"TEST NOT EXECUTED"
077574  000402                          BR      100$                    ;JUST EXIT IF NOT
077576  012700  101222          1$:     MOV     @TS140ID,R0             ;TEST ID MESSAGE
077602  004737  016510          100$:   JSR     PC,TSTSETUP             ;DO THE COMMON SETUP
077606  004737  017762                  JSR     PC,CHKMAN               ;SEE IF MANUAL INTERVENTION ALLOWED
077612  103402                          BCS     2$                      ;CARRY SET IF INTERVENTION ALLOWED
077614  000137  100276                  JMP     64$                     ;EXIT IF NO MANUAL INTERVENTION
077620  004737  015774          2$:     JSR     PC,SOFINIT              ;DO A SOFT INIT
077624  103405                          BCS     5$                      ;BRANCH IF OK
077626  010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
077630                                  ERROF   ERRNU,SFIERR,SFIMSG     ;REPORT FATAL ERROR
077630  104455                                                                  TRAP    C$ERDF
077632  002261                                                                  .WORD   1201
077634  003652                                                                  .WORD   SFIERR
077636  012034                                                                  .WORD   SFIMSG
077640  012700  100310          5$:     MOV     @SCMENU,R0              ;MENU OF SCOPE LOOP SELECTIONS
```

TEST 12: SCOPE LOOPS

```
10886 077644  012701  000007                MOV     #7.,R1                  ;MAXIMUM ALLOWED SELECTION
10887 077650  004737  017540                JSR     PC,GETSEL               ;GO GET THE OPERATORS SELECTION
10888 077654  005700                       TST     R0                      ;WAS ZERO SPECIFIED ?
10889 077656  001760                       BEQ     2$                      ;REPEAT MENU IF YES.
10890 077660  020027  000007                CMP     R0,#7                   ;EXTENDED TSSR ?
10891 077664  001015                       BNE     3$                      ;BRANCH IF NOT
10892 077666  005737  002226                TST     EXTFEA                          ;CHECK FOR EXTENDED FEATURES SET
10893 077672  001012                       BNE     3$                      ;BR, IF IT IS ON
10894 077674                               PRINTF  #EXFMSG                 ;WARN OPERATOR EXTENDED FEATURES CLEAR
      077674  012746  101077                                                          MOV     #EXFMSG,-(SP)
      077700  012746  000001                                                          MOV     #1,-(SP)
      077704  010600                                                                  MOV     SP,R0
      077706  104417                                                                  TRAP    C$PNTF
      077710  062706  000004                                                          ADD     #4,SP
10895 077714  000137  077620                JMP     2$                      ;GO BACK TO BASIC MENU
10896 077720  010004                3$:     MOV     R0,R4                   ;SAVE THE MENU SELECTION
10897 077722                               SETPRI  #PRI07                  ;RAISE THE PRIORITY
      077722  012700  000340                                                          MOV     #PRI07,R0
      077726  104441                                                                  TRAP    C$SPRI
10898 077730  005037  100302                CLR     TTION                   ;ASSUME INTERRUPTS ARE ENABLED
10899 077734  032737  000100  177560        BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
10900 077742  001005                       BNE     4$                      ;BRANCH IF YES
10901 077744  005237  100302                INC     TTION                   ;FLAG SET IF INTERRUPTS OFF
10902 077750  052737  000100  177560        BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
10903 077756  012701  000060                4$:     MOV     #TTIVEC,R1      ;START OF TTI VECTORS
10904 077762  011137  100304                MOV     (R1),TVECSAV            ;SAVE THE CURRENT TTI VECTOR
10905 077766  012721  100210                MOV     #60$,(R1)+              ;SET NEW INTERRUPT ROUTINE
10906 077772  011137  100306                MOV     (R1),TPRISAV            ;SAVE THE VECTOR PRIORITY
10907 077776  012711  000340                MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
10908 100002                               SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
      100002  012700  000000                                                          MOV     #PRI00,R0
      100006  104441                                                                  TRAP    C$SPRI
10909 100010  006304                       ASL     R4                      ;CONVERT TO WORD OFFSET
10910 100012  000174  100016                JMP     @6$(R4)                 ;JUMP TO PROPER LOOP
10911 100016  077620                6$:     .WORD   2$                      ;RETYPE THE MENU
10912 100020  100036                        .WORD   10$                     ;TSBA READ ACCESS
10913 100022  100046                        .WORD   15$                     ;TSSR READ ACCESS
10914 100024  100060                        .WORD   20$                     ;TSSR WRITE ACCESS
10915 100026  100100                        .WORD   25$                     ;TSDB HIGH BYTE WRITE ACCESS
10916 100030  100124                        .WORD   30$                     ;TSDB LOW BYTE WRITE ACCESS
10917 100032  100150                        .WORD   35$                     ;TSDB MAINTENANCE MODE
10918 100034  100170                        .WORD   40$                     ;TSDBX WRITE ACCESS
10919                               ;                                       ;LEAVE THE TEST
10920
10921
10922 100036  105065  000000        10$:    CLRB    TSDB(R5)                ;ENTER MAINTENANCE MODE
10923 100042  011500                12$:    MOV     (R5),R0                 ;READ TSBA REGISTER
10924 100044  000776                        BR      12$                     ;LOOP UNTIL HALTED
10925
10926
10927 100046  012703  000002        15$:    MOV     #TSSR,R3                ;ADDRESS OF TSSR REGISTER
10928 100052  060503                        ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
10929 100054  011300                18$:    MOV     (R3),R0                 ;READ TSSR REGISTER
10930 100056  000776                        BR      18$                     ;LOOP UNTIL STOPPED
10931
10932 100060  004737  017456        20$:    JSR     PC,GETPAT               ;READ THE DATA PATTERN
10933 100064  010001                        MOV     R0,R1                   ;DATA PATTERN FOR LOOP
```

TEST 12: SCOPE LOOPS

```
10934 100066  012703  000002            MOV     #TSSR,R3                ;ADDRESS OF TSSR
10935 100072  060503                    ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
10936 100074  010113              22$:  MOV     R1,(R3)                 ;WRITE DATA TO TSSR
10937 100076  000776                    BR      22$                     ;LOOP
10938
10939
10940 100100  105065  000000      25$:  CLRB    TSDB(R5)                ;ENTER MAINTENANCE MODE
10941 100104  004737  017456            JSR     PC,GETPAT               ;READ THE DATA PATTERN
10942 100110  010001                    MOV     R0,R1                   ;DATA PATTERN FOR LOOP
10943 100112  012703  000001            MOV     #TSDBH,R3               ;ADDRESS OF HIGH BYTE OF TSDB
10944 100116  060503                    ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
10945 100120  110113              27$:  MOVB    R1,(R3)                 ;WRITE THE DATA TO TSDB, HIGH BYTE
10946 100122  000776                    BR      27$                     ;LOOP UNTIL STOPPED
10947
10948
10949 100124  105065  000000      30$:  CLRB    TSDB(R5)                ;ENTER MAINTENANCE MODE
10950 100130  004737  017456            JSR     PC,GETPAT               ;READ THE DATA PATTERN
10951 100134  010001                    MOV     R0,R1                   ;DATA PATTERN FOR LOOP
10952 100136  012703  000000            MOV     #TSDB,R3                ;ADDRESS OF TSSR
10953 100142  060503                    ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
10954 100144  110113              32$:  MOVB    R1,(R3)                 ;WRITE DATA TO TSSR, LOW BYTE
10955 100146  000776                    BR      32$                     ;LOOP UNTIL HALTED BY OPERATOR
10956
10957 100150  004737  017456      35$:  JSR     PC,GETPAT               ;READ THE DATA PATTERN
10958 100154  010001                    MOV     R0,R1                   ;DATA PATTERN FOR LOOP
10959 100156  012703  000000            MOV     #TSDB,R3                ;SELECT TSDB
10960 100162  060503                    ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
10961 100164  010113              37$:  MOV     R1,(R3)                 ;WRITE THE DATA PATTERN
10962
10963 100166  000776                    BR      37$                     ;LOOP UNTIL HALTED
10964
10965 100170  004737  017456      40$:  JSR     PC,GETPAT               ;READ THE DATA PATTERN
10966 100174  010001                    MOV     R0,R1                   ;SAVE THE DATA PATTERN
10967 100176  012703  000003            MOV     #TSSRH,R3               ;BYTE ADDRESS OF TSSR, HIGH BYTE
10968 100202  060503                    ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
10969 100204  110113              42$:  MOVB    R1,(R3)                 ;WRITE THE DATA TO REGISTER
10970 100206  000776                    BR      42$                     ;LOOP UNTIL HALTED
10971
10972
10973                             ;+
10974                             ;PROCESS CONSOLE INTERRUPTS
10975                             ;-
10976
10977 100210  010046              60$:  MOV     R0,-(SP)                ;SAVE WORK REGISTER
10978 100212  113700  177562            MOVB    @#TTIBFR,R0             ;GET THE OPERATOR INPUT
10979 100216  042700  000200            BIC     #200,R0                 ;STRIP OFF PARITY BIT
10980 100222  122700  000015            CMPB    #15,R0                  ;IS IT A CARRIAGE RETURN ?
10981 100226  001021                    BNE     61$                     ;JUST EXIT IF NOT
10982 100230  012766  077620  000002    MOV     #2$,2(SP)               ;RETURN TO MASTER MENU
10983 100236  005066  000004            CLR     4(SP)                   ;FORCE PRIORITY ZERO
10984 100242  013737  100304  000060    MOV     TVECSAV,@#TTIVEC        ;RESTORE SUPERVISOR VECTOR
10985 100250  013737  100306  000062    MOV     TPRISAV,@#TTIVEC+2      ;RESTORE SUPERVISOR PRIORITY
10986 100256  005737  100302            TST     TTION                   ;ARE SUPERVISOR INTERRUPTS ENABLED ?
10987 100262  001403                    BEQ     61$                     ;BRANCH IF YES
10988 100264  042737  000100  177560    BIC     #100,@#TTICSR           ;TURN OFF TTI INTERRUPTS
10989 100272  012600              61$:  MOV     (SP)+,R0                ;RESTORE REGISTER
10990 100274  000002                    RTI                             ;RETURN FROM INTERUPT
```

TEST 12: SCOPE LOOPS

```
10991
10992 100276                          64$:
10993 100276                          63$:    EXIT    TST                     ;EXIT THE TEST
      100276  104432                                                                TRAP    C$EXIT
      100300  000736                                                                .WORD   L10077-.
10994                                 ;65$:   JMP     200                     ;RETURN TO SUPERVISOR
10995                                 ;**************JUMP TO 200 IS NOT SUPPORTED ON XXDP+ VERSION 2.X.
10996                                 ;+
10997                                 ;LOCAL STORAGE FOR THIS TEST
10998                                 ;-
10999
11000 100302  000000                  TTION:          .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
11001 100304  000000                  TVECSAV:        .WORD   0               ;SAVE TTI VECTOR
11002 100306  000000                  TPRISAV:        .WORD   0               ;SAVE TTI PRIORITY
11003
11004
11005                                 ;+
11006                                 ;MENU FOR OPERATOR INPUT FOR SCOPE LOOPS
11007                                 ;-
11008
11009                                         .EVEN
11010 100310  100342  100415  100443  SCMENU: .WORD   1$,2$,3$,4$,5$,6$
11011 100324  100614  100652  100720          .WORD   7$,8$,9$,10$,11$,12$,0
11012
11013
11014 100342     012     123     105  1$:     .ASCIZ  <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
11015 100415     012     011     060  2$:     .ASCIZ  <12>'   0       Display This Menu'
11016 100443     011     061     011  3$:     .ASCIZ  '       1       TSBA Read Access'
11017 100467     011     062     011  4$:     .ASCIZ  '       2       TSSR Read Access'
11018 100513     011     063     011  5$:     .ASCIZ  '       3       Initialize (TSSR Write Access)'
11019 100555     011     064     011  6$:     .ASCIZ  '       4       TSDB High Byte Write Access'
11020 100614     011     065     011  7$:     .ASCIZ  '       5       TSDB Low Byte Write Access'
11021 100652     011     066     011  8$:     .ASCIZ  '       6       TSDB Maintenance Mode Write Access'
11022 100720     011     067     011  9$:     .ASCIZ  '       7       TSDBX (TSSR High Byte) Write Access'
11023 100767     011     136     103  10$:    .ASCIZ  '       ^C      To Return to Diagnostic Supervisor'
11024 101036     000             000  11$:    .ASCIZ  ''
11025 101037     124     171     160  12$:    .ASCIZ  'Type RETURN To Stop Scope Loops'
11026 101077     045     116     045  EXFMSG: .ASCIZ  '%N%A *** Extended Features Switch Not On *** '
11027 101155     123     164     141  T4ONE:  .ASCIZ  'Stand-alone Scope Loops Not Executed'
11028 101222     123     143     157  TST4OID:        .ASCIZ  'Scope Loops'
11029                                         .EVEN
11030 101236                                  ENDTST
      101236                                                                    L10077:
      101236  104401                                                                TRAP    C$ETST
11031 101240                                  ENDMOD
11033                                         .TITLE  TSV6 - PARAMETER CODING
11039
11044
11050
11051 101240                                  BGNMOD  TSV6
      101240                          TSV6::
11052
11053                                         .SBTTL  HARDWARE PARAMETER CODING SECTION
11054
11055                                 ;++
11056                                 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
11057                                 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
```

HARDWARE PARAMETER CODING SECTION

```
11058                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
11059                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
11060                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
11061                                   ; WITH THE OPERATOR.
11062                                   ;--
11063 101240                                    BGNHRD
      101240  000010                            .WORD L10100-L$HARD/2
      101242                            L$HARD::
11064
11065 101242                                    GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
      101242  000031                            .WORD   T$CODE
      101244  101262                            .WORD   HPM1
      101246  160010                            .WORD   T$LOLIM
      101250  177776                            .WORD   T$HILIM
11066 101252                                    GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
      101252  001031                            .WORD   T$CODE
      101254  101316                            .WORD   HPM2
      101256  000000                            .WORD   T$LOLIM
      101260  C00776                            .WORD   T$HILIM
11067                                           ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
11068 101262                                    ENDHRD
                                                .EVEN
      101262                            L10100:
11069 101262     104     105     126    HPM1:   .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB, '
11070 101316     111     116     124    HPM2:   .ASCIZ  'INTERRUPT VECTOR   '
11071 101342     111     116     124    HPM3:   .ASCIZ  'INTERRUPT PRIORITY     '
11072                                           .EVEN
```

SOFTWARE PARAMETER CODING SECTION

```
                                        .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--
101372                                  BGNSFT
101372  000003                          .WORD L10101-L$SOFT/2
101374                          L$SOFT::
                                ;       GPRML   SPM1,0,-1,YES                   ; GET TRANSPORT TEST FLAG.
101374                                  GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
101374  001130                          .WORD   T$CODE
101376  101432                          .WORD   SPM4
101400  177777                          .WORD   -1
                                ;       GPRMD   SPM6,4,D,7777,0,7777,YES        ; GET LOCAL ERROR LIMIT
                                ;       GPRMD   SPM7,6,D,7777,0,7777,YES        ; GET GLOBAL ERROR LIMIT
101402                                  ENDSFT
                                        .EVEN
101402                          L10101:

101402  105  116  101  SPM1:    .ASCIZ  'ENABLE TRANSPORT TESTS '
101432  111  116  110  SPM4:    .ASCIZ  'INHIBIT ITERATIONS     '
101462  120  105  122  SPM6:    .ASCIZ  'PER TEST ERROR LIMIT   '
101512  120  105  122  SPM7:    .ASCIZ  'PER UNIT ERROR LIMIT   '
                                        .SBTTL  PATCH AREA


                                ;
                                ; FINALLY A GENEROUS PATCH AREA.
                                ;
                                ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
                                ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
                                ;

101542                          PATCH::

101542                                  .BLKW   32.

                                        .=.!377+1
102000                                  LASTAD          ;SET LAST USED ADDRESS.
                                        .EVEN
102000  000000                          .WORD   0
102002  000000                          .WORD   0
102004                          L$LAST::
                                        ENDMOD
000001                                  .END
```

Symbol table

```
ADDSSR  012126 G        C$AUTO= 000061        DEVNRD  022645        FUSI    004117        INTMAS  016144
ADR   = 000020 G        C$BRK = 000022        DEVNXR  022563        F$AU  = 000015        INTR    016216 G
AMBTSS  006635          C$BSEG= 000004        DEVONL  022513        F$AUTO= 000020        INTREC  002224 G
ASSEMB= 000010          C$BSUB= 000002        DEVSUM  022456        F$BGN = 000040        INTVEC  016146
A1716 = 000003          C$CEFG= 000045        DFPTBL  002156 G      F$CLEA= 000007        INTX    004300
BADDAT  003156 G        C$CLCK= 000062        DIAGMC= 000000        F$DU  = 000016        INVERT  020444 G
BADSSR  015700 G        C$CLEA= 000012        DICEB = 000001        F$END = 000041        IOKCKI= 000200
BDVPCR= 177520 G        C$CLOS= 000035        DSBINT  016204        F$HARD= 000004        IOKSTP= 000001
BENBSW  002230 G        C$CLP1= 000006        DUAD12  004643        F$HW  = 000013        IPRI    002212 G
BIE   = 040000          C$CVEC= 000036        DUFLG   003112 G      F$INIT= 000006        ISR   = 000100 G
BIT0  = 000001 G        C$DCLN= 000044        DUMMY   003062        F$JMP = 000050        IVEC    002210 G
BIT00 = 000001 G        C$DODU= 000051        EF.CON= 000036 G      F$MOD = 000000        IXE   = 004000 G
BIT01 = 000002 G        C$DRPT= 000024        EF.NEW= 000035 G      F$MSG = 000011        I$AU  = 000041
BIT02 = 000004 G        C$DU  = 000053        EF.PWR= 000034 G      F$PROT= 000021        I$AUTO= 000041
BIT03 = 000010 G        C$EDIT= 000003        EF.RES= 000037 G      F$PWR = 000017        I$CLN = 000041
BIT04 = 000020 G        C$ERDF= 000055        EF.STA= 000040 G      F$RPT = 000012        I$DU  = 000041
BIT05 = 000040 G        C$ERHR= 000056        EMAXDU  016777        F$SEG = 000003        I$HRD = 000041
BIT06 = 000100 G        C$ERRO= 000060        EN    = 000000        F$SOFT= 000005        I$INIT= 000041
BIT07 = 000200 G        C$ERSF= 000054        ENAINT  016152        F$SRV = 000010        I$MOD = 000041
BIT08 = 000400 G        C$ERSO= 000057        ENVIRN  020112        F$SUB = 000002        I$MSG = 000041
BIT09 = 001000 G        C$ESCA= 000010        EPRTSW  002200 G      F$SW  = 000014        I$PROT= 000040
BIT1  = 000002 G        C$ESEG= 000005        EPRT1   006360        F$TEST= 000001        I$PTAB= 000041
BIT10 = 002000 G        C$ESUB= 000003        EPRT2   006360        GDDAT   003160 G      I$PWR = 000041
BIT11 = 004000 G        C$ETST= 000001        ERCM    011733        GERRMA  002174 G      I$RPT = 000041
BIT12 = 010000 G        C$EXIT= 000032        ERRHI   002236 G      GETPAT  017456 G      I$SEG = 000041
BIT13 = 020000 G        C$GETB= 000026        ERRK    016756        GETSEL  017540 G      I$SETU= 000041
BIT14 = 040000 G        C$GETW= 000027        ERRLO   002240 G      G$CNTO= 000200        I$SFT = 000041
BIT15 = 100000 G        C$GMAN= 000043        ERRNO = 002261        G$DELM= 000372        I$SRV = 000041
BIT2  = 000004 G        C$GPHR= 000042        ERRVEC= 000004 G      G$DISP= 000003        I$SUB = 000041
BIT3  = 000010 G        C$GPLO= 000030        ERTABE  003376        G$EXCP= 000400        I$TST = 000041
BIT4  = 000020 G        C$GPRI= 000040        ERTABL  003176        G$HILI= 000002        J$JMP = 000167
BIT5  = 000040 G        C$INIT= 000011        ESUM    016760        G$LOLI= 000001        KIPAR0= 172340
BIT6  = 000100 G        C$INLP= 000020        EVL   = 000004 G      G$NO  = 000000        KIPAR1= 172342
BIT7  = 000200 G        C$MANI= 000050        EXBCNT= 000010        G$OFFS= 000400        KIPAR2= 172344
BIT8  = 000400 G        C$MEM = 000031        EXFMSG  101077        G$OFSI= 000376        KIPAR3= 172346
BIT9  = 001000 G        C$MSG = 000023        EXPBRE  015502 G      G$PRMA= 000001        KIPAR4= 172350
BOE   = 000400 G        C$OPEN= 000034        EXPD    002232 G      G$PRMD= 000002        KIPAR5= 172352
BRINIT  004457          C$PNTB= 000014        EXPGOT  004533        G$PRML= 000000        KIPAR6= 172354
BSEL0 = 000000          C$PNTF= 000017        EXPGT2  004567        G$RADA= 000140        KIPAR7= 172356
BSEL1 = 000001          C$PNTS= 000016        EXPMSG  002322 G      G$RADB= 000000        KIPDR0= 172300
CHKAMB  016044          C$PNTX= 000015        EXPREC  015474 G      G$RADD= 000040        KIPDR1= 172302
CHKMAN  017762 G        C$QIO = 000377        EXTA    005772        G$RADL= 000120        KIPDR2= 172304
CHKTSS  016336          C$RDBU= 000007        EXTEND  005770        G$RADO= 000020        KIPDR3= 172306
CKDROP  017202          C$REFG= 000047        EXTFEA  002226 G      G$XFER= 000004        KIPDR4= 172310
CKEMAX  017102          C$RESE= 000033        E$END = 002100        G$YES = 000010        KIPDR5= 172312
CKMSG   011360 G        C$REVI= 000003        E$LOAD= 000035        HIADDR= 001400        KIPDR6= 172314
CKMSG2  011500 G        C$RFLA= 000021        FATERR= 000060        HOE   = 100000 G      KIPDR7= 172316
CKRAM   011114 G        C$RPT = 000025        FATFLG  002222 G      HPM1    101262        KTENAB  003134 G
CKRAM2  011224 G        C$SEFG= 000046        FERCM   011722        HPM2    101316        KTFLG   003132 G
CMDPKT  020610 G        C$SPRI= 000041        FIFEXP  012170 G      HPM3    101342        KTINIT  020272
CONFIG  017250          C$SVEC= 000037        FIF1MS  012242        IBE   = 010000 G      KTOFF   017274
COUNT   002310 G        C$TPRI= 000013        FIF2MS  012311        IDU   = 000040 G      KTON    017256
CSRADD  002206 G        DATA    002312 G      FNOINT  004215        IER   = 020000 G      LERRMA  002172 G
CTAB    003164 G        DATASC  017514        FORCER  002176 G      IFAULT  004256        LERRNO= 000000
CTABE   003176 G        DEBUGM  011632        FREE    003124 G      INCERK  017044        LISTAL= 000001
CTABM   003164 G        DEVCNT  002220 G      FREEHI  003130        INTCPC  016150        LOE   = 040000 G
C$AU  = 000052          DEVDRO  022726        FRESIZ  003126 G      INTFLA  016145        LOOPCN  002216 G
```

Symbol table

```
LOOPCO  013126          L10001  002176          L10073  066214          ONEFIL= 000000          PRMSGE  014552 G
LOOPFL  003162 G        L10002  005766          L10074  064354          O$APTS= 000000          PRMSGO  014732
LOT   = 000010 G        L10003  012044          L10075  074052          O$AU  = 000001          PRMSG1  014777
L$ACP   002110 G        L10004  012062          L10076  077562          O$BGNR= 000001          PRMSG2  015035
L$APT   002036 G        L10005  012100          L10077  101236          O$BGNS= 000001          PROASC  014420
L$AU    021702 G        L10006  012106          L10100  101262          O$DU  = 000001          PR1ASC  014465
L$AUT   002070 G        L10007  012124          L10101  101402          O$ERRT= 000000          PST32W  003152 G
L$AUTO  022106 G        L10010  012142          MEMADD  013754 G        O$GNSW= 000001          PUNIT   021634
L$CCP   002106 G        L10011  012166          MEMCK   020626 G        O$POIN= 000001          PW.D11= 000021
L$CLEA  022166 G        L10012  012240          MENASC  017731          O$SETU= 000000          PW.D13= 000022
L$CO    002032 G        L10013  012410          MENERR  017656          PASRPT  021404          PW.D22= 000020
L$DEPO  002011 G        L10014  013124          MENRES  017760          PATCH   101542 G        PW.NOP= 000000
L$DESC  003410 G        L10015  013752          MIMENU  072732          PATDAT  017512          PW.NO1= 000023
L$DESP  002076 G        L10016  013774          MMVEC = 000250          PC.ERA= 002400          PW.RDE= 000024
L$DEVP  002060 G        L10017  015500          MSA.FR= 000006          PC.IER= 002000          PW.RDR= 000001
L$DISP  002124 G        L10020  015506          MSA.NO= 000000          PC.NOO= 001000          PW.RDS= 000005
L$DLY   002116 G        L10021  015514          MSA.NR= 000004          PC.REL= 000000          PW.RFI= 000003
L$DTP   002040 G        L10022  015526          MSA.VO= 000002          PC.REW= 000400          PW.WCT= 000006
L$DTYP  002034 G        L10023  015550          MSGEXP  012144 G        PKBCNT= 000006          PW.WFI= 000004
L$DU    022000 G        L10024  015576          MSGLOO  013064 G        PKHI  = 000004          PW.WFM= 000007
L$DUT   002072 G        L10025  015736          MSGSTA  012350 G        PKLOW = 000002          PW.WMI= 000010
L$DVTY  003402 G        L10026  016246          MSGSUB  013742 G        PKTADD  007554          PW.WNP= 000011
L$EF    002052 G        L10030  021632          MS.ATT= 000006          PKTFRM  007516          PW.WTR= 000002
L$ENVI  002044 G        L10031  021776          MS.EXT= 000200          PKTGET  012064 G        P.ACK = 100000
L$ETP   002102 G        L10032  022104          MS.RSD= 000001          PKTMES  012110 G        P.CMD = 000037
L$EXP1  002046 G        L10033  022164          MS.RSF= 000020          PKTRAM  004745 G        P.CONT= 000012
L$EXP4  002064 G        L10034  022212          MS.RST= 000010          PKTSSR  012046 G        P.CVC = 040000
L$EXP5  002066 G        L10035  022454          M8186   005554          PNT   = 001000 G        P.FMT = 000140
L$HARD  101242 G        L10036  023754          M8189   005645          PRAMPK  013776          P.FORM= 000011
L$HIME  002120 G        L10037  025746          NBA   = 002000          PRASC   014523          P.GETS= 000017
L$HPCP  002016 G        L10040  024230          NEWPAS  021340          PRBEXP  015470          P.IE  = 000200
L$HPTP  002022 G        L10041  024474          NODEV   003114 G        PRBMSG  015336          P.INIT= 000013
L$HW    002156 G        L10042  031342          NOEXTF  027536          PRBREC  015472          P.MODE= 007400
L$ICP   002104 G        L10043  026322          NOINIT  004335          PRBTOT  015423          P.OPP = 020000
L$INIT  021106 G        L10044  026612          NOINTR  004221          PRBYTE  015122 G        P.POSI= 000010
L$LADP  002026 G        L10045  027110          NOITS   002170 G        PRI   = 002000 G        P.READ= 000001
L$LAST  102004 G        L10046  027542          NOMAN   020016          PRIADD  010160          P.SWB = 010000
L$LOAD  002100 G        L10047  034132          NOMEM   005460          PRIAO   010230          P.WRIT= 000005
L$LUN   002074 G        L10050  031676          NP.IR = 000200          PRIBXO  007612 G        P.WRTC= 000004
L$MREV  002050 G        L10051  032270          NP.LOO= 000040          PRIEQU  010060          P.WRTS= 000006
L$NAME  002000 G        L10052  032676          NP.OUT= 000100          PRIPKT  007370 G        QVP     002204 G
L$PRIO  002042 G        L10053  037724          NP.WRP= 000020          PRIRAM  010066          RAMASC  014156
L$PROT  021076 G        L10054  035216          NSI     004152          PRITAD  010274          RAMDAT  002242 G
L$PRT   002112 G        L10055  036150          NSINIT  004407          PRITSS  006024          RAMERR  015510 G
L$REPP  002062 G        L10056  050036          NUL     004527          PRITO   010356          RAMEXP  015530 G
L$REV   002010 G        L10057  040230          NULCR   004530          PRIT1   010421          RAMFOR  010116
L$RPT   022214 G        L10060  041440          NUMENU  075342          PRIXOR  007742 G        RAMSIZ  002302 G
L$SOFT  101374 G        L10061  043000          NXM   = 004000          PRIO0 = 000000 G        RAMTAD  015516 G
L$SPC   002056 G        L10062  043356          NXMFLG  003136 G        PRIO1 = 000040 G        RCVHIA  002304 G
L$SPCP  002020 G        L10063  044630          NXMHI   003142 G        PRIO2 = 000100 G        RCVLOA  002306 G
L$SPTP  002024 G        L10064  045674          NXMLO   003140 G        PRIO3 = 000140 G        RDERR   005206
L$STA   002030 G        L10065  051316          NXMTST  021002          PRIO4 = 000200 G        RECMSG  002466 G
L$SW    002166 G        L10066  062144          NXR     003740          PRIO5 = 000240 G        RECV    002234 G
L$TEST  002114 G        L10067  053322          NXRERR  005736 G        PRIO6 = 000300 G        REGSAV  017422
L$TIML  002014 G        L10070  054610          NXRX    003777          PRIO7 = 000340 G        RETERR  005372
L$UNIT  002012 G        L10071  056060          NXTU    021352          PRMESS  014242          REWIND  011014 G
L10000  002164          L10072  057324          OFL   = 000100          PRMNO   002320 G        RMCHBE= 000167
```

Symbol table

```
RMCHEN= 000200          S1.I1R= 020000          TSV2    002000 G        T11     074054 G        T15BF2  033430
RMMSGB= 000215          S1.I2R= 040000          TSV3    002176 G        T12     077564 G        T15BS0  033430
RMMSGE= 000234          S1.PAR= 100000          TSV4    021076 G        T12BFR  027632          T15BS1  033431
RMPKTB= 000201          S2.ATI= 000010          TSV5    022776 G        T12BKG  030307          T15DAT  032730
RMPKTE= 000210          S2.BTI= 000004          TSV6    101240 G        T12BLK  027664          T15L00  031374
RMR   = 010000          S2.DIM= 000200          TTIBFR= 177562 G        T12CHA  031300          T15PAC  032720
RWPACK  011110          S2.ILW= 000100          TTICSR= 177560 G        T12CKR  031060 G        T15PK2  033420
SC    = 100000          S2.INR= 000020          TTION   100302          T12CON  030666          T15RES  034022
SCE   = 020000          S2.OUT= 000040          TTION2  070776          T12DAT  027620          T15RT2  034074
SCHERR  005300          S2.UND= 000003          TTIVEC= 000060 G        T12DPR  030466          T15SSR  033436
SCME    005013          TBLEND= 003062 G        TVECSA  100304          T12GET  030046          T15S2   033432
SCMENU  100310          TCOASC  006476          TVSAV2  071000          T12HIA  027652          T15S3   033434
SDELAY  010660          TCOCOD  006676          T$ARGC= 000001          T12KT   027660          T16BEN  037610
SELASC  017724          TEMP1   003116 G        T$CODE= 001130          T12LOA  027654          T16BFR  037562
SELDAT= 000004          TEMP2   003120 G        T$ERRN= 002261          T12L00  025776          T16BFS  037602
SEL2  = 000002          TERCLS= 000016          T$EXCP= 000000          T12MSG  030211          T16CLR  037426
SETMAP  017316          TESTNO= 000014          T$FLAG= 000040          T12NIN  030375          T16DAT  037550
SETU    021436          TEXASC  006435          T$GMAN= 000000          T12NXM  030557          T16DT2  037620
SFFMSG  012102 G        TFCASC  006537          T$HILI= 000776          T12PAC  027610          T16D01  036202
SFHERR  003705          TIMEXP  015552 G        T$LAST= 000001          T12PAR  027656          T16D28  036204
SFIERR  003652          TIMSG0  015600          T$LOLI= 000000          T12SET  031236          T16D53  036206
SFIMSG  012034 G        TINERR  012021          T$LSYM= 010000          T12SWR  031170          T16D78  036210
SFPTBL  002166 G        TMPBFR  002632 G        T$LTNO= 000014          T12TBE  030016          T16LEN  036532
SIFLAG  003154 G        TNAM    016704          T$NEST= 177777          T12WRT  030122          T16L00  034152
SIMSG   011766          TPRISA  100306          T$NS0 = 000000          T121L0  026110          T16PAC  037540
SKIPT   003400          TPSAV2  071002          T$NS1 = 000005          T122L0  026444          T16PK2  037610
SOFINI  015774 G        TRANST  002166 G        T$NS2 = 000002          T123L0  026664          T16REJ  036644
SPACE   010466 G        TSBA  = 000000 G        T$NS3 = 000003          T124L0  027306          T16RES  037360
SPM1    101402          TSBAH = 000001 G        T$PTNU= 000000          T124TS  027662          T16SEX  037520
SPM4    101432          TSDB  = 000000 G        T$SAVL= 177777          T13BFR  023412          T16SRD  037452
SPM6    101462          TSDBH = 000001 G        T$SEGL= 177777          T13DAT  023400          T16SSR  036262
SPM7    101512          TSFCOD  007236          T$SEKO= 010000          T13L00  023014          T16TSB  036430
SR0   = 177572          TSREJ = 000006          T$SUBN= 000000          T13MEM  023505          T16T01  036761
SR1   = 177574          TSSDEF  006606          T$TAGL= 177777          T13NBA  023544          T16T28  037060
SR2   = 177576          TSSR  = 000002 G        T$TAGN= 010102          T13PAC  023370          T16T53  037160
SR3   = 172516          TSSRBI  003502 G        T$TEMP= 000000          T13RES  023706          T16T78  037260
SSR   = 000200          TSSRFO  006415          T$TEST= 000014          T13SSR  023617          T16WMI  037472
STATCO  012412          TSSRH = 000003 G        T$TSTM= 177777          T14BFR  024526          T162SS  036317
SVCGBL= 000000          TSSX    004020          T$TSTS= 000001          T14BS0  024520          T163SS  036363
SVCINS= 000000          TSTBLK  002752 G        T$$AU = 010031          T14BS1  024521          T17BEN  047712
SVCSUB= 000001          TSTCNT  002214 G        T$$AUT= 010033          T14BS2  024522          T17BFR  047572
SVCTAG= 000000          TSTEND  016720          T$$CLE= 010034          T14DAT  024520          T17BFS  047612
SVCTST= 000001          TSTFLA  002314 G        T$$DU = 010032          T14DTA  025140          T17CLE  047476
S$LSYM= 010000          TSTL00  016456 G        T$$HAR= 010100          T14L00  023774          T17CLR  047310
S0.IDB= 000010          TSTPTR  002316 G        T$$HW = 010000          T14NBA  025152          T17DAT  047560
S0.IFB= 000002          TSTSET  016510 G        T$$INI= 010030          T14NIN  025413          T17DT2  047730
S0.IFP= 000001          TST12I  030020          T$$MSG= 010025          T14PAC  024510          T17EXE  046054
S0.ILD= 000020          TST13I  023432          T$$PRO= 010027          T14PK2  025130          T17EXP  045732
S0.ION= 000040          TST14I  025571          T$$RPT= 010035          T14RES  025636          T17EXS  045752
S0.IRD= 000100          TST15I  034003          T$$SEG= 010000          T14RST  025674          T17L00  037754
S0.IRW= 000004          TST16I  036212          T$$SOF= 010101          T14SSR  025323          T17MSK  045726
S0.ISP= 000200          TST17I  046156          T$$SRV= 010026          T14TSB  025505          T17PAC  047550
S1.ICE= 002000          TST18I  050546          T$$SUB= 010074          T142RE  025226          T17PK2  047720
S1.IEO= 010000          TST19I  057532          T$$SW = 010001          T15AM2  033512          T17RFI  047456
S1.IFM= 001000          TST20I  064532          T$$TES= 010077          T15AM3  033613          T17RSF  047354
S1.IHE= 000400          TST39I  077534          T1      022776 G        T15AM4  033715          T17SET  047516
S1.IID= 004000          TST40I  101222          T10     066216 G        T15BFR  032742          T17SNP  047376
```

Symbol table

```
T17SRD  047334          T19WCT  061472          T38FLG  003150 G        T39NE   076333          WF.IED= 000010
T17SSR  046175          T19WFI  061436          T3.1    025776          T39NFL  076330          WF.IER= 000004
T17WFD  046054          T19WFM  061512          T3.2    026336          T39OFF  077376          WF.IHI= 000200
T17WFI  047422          T19WST  061073          T3.3    026626          T39OF2  076574          WF.IRE= 000040
T171CM  046515          T191CM  060230          T3.4    027124          T39ON   077405          WF.IWF= 000020
T172CM  046577          T192CM  060312          T38ASC  074005          T39ON2  076640          WF.IWR= 000100
T172SS  046232          T192SS  057630          T38ASN  074024          T39OUT  077201          WF.I3R= 000002
T173CM  046673          T193CM  060406          T38ASO  073702          T39PAC  075560          WF.I4R= 000001
T173SS  046276          T193SS  057674          T38AS1  073747          T39PK2  076260          WRTCMR  010662 G
T174CM  046757          T194SS  057741          T38BFR  071026          T39PK3  076310          WRTERR  005113
T174SS  046343          T195CM  060474          T38BSO  071020          T39PK4  076320          WRTMSG  005056
T175CM  047042          T195SS  060004          T38BS1  071021          T39RES  077476          WSMBK   020620 G
T175SS  046406          T196CM  060550          T38BS2  071022          T39RL   077472          XFERAS  015740
T176CM  047116          T196SS  060050          T38CNT  074050          T39SBS  077324          XNXM    016376
T176SS  046452          T197CM  060633          T38DAT  073340          T39SFS  077252          XORBFO  007674
T177CM  047224          T197SS  060113          T38DLY  071004          T39SIZ  076326          XORFOR  010012
T188FR  051252          T198CM  060721          T38DSW  071530          T39SSR  077111          XSTO  = 000006 G
T18CMP  050753          T198SS  060162          T38DTA  071520          T39TAD  075570          XST1  = 000010 G
T18DAT  051240          T199CM  061010          T38EAI  071526          T39WPN  076756          XST2  = 000012 G
T18DT2  051310          T2      023756 G        T38EB   071502          T39WR   076322          XST3  = 000014 G
T18EXP  050516          T2.1    023774          T38ID   072705          T39WRT  076703          XST4  = 000016 G
T18LOO  050056          T2.2    024244          T38INT  072276          T4      031344 G        XSOBOT= 000002
T18MSK  050512          T20BEN  066072          T38MBP  073400          T4.1    031374          XSOEOT= 000001
T18PAC  051230          T20BFR  065752          T38MSG  072645          T4.2    031700          XSOIE = 000040
T18PK2  051300          T20BFS  065772          T38MS1  072506          T4.3    032272          XSOILA= 000400
T18SET  051120          T20CLE  065664          T38MS2  072601          T40NE   101155          XSOILC= 001000
T18SMI  051076          T20CLR  065452          T38MS3  071645          T5      034134 G        XSOLET= 020000
T18SRD  051056          T20CWP  065357          T38NBA  072132          T5.1    034152          XSOMOT= 000200
T18SSR  050605          T20DAT  065740          T38NE   071570          T5.2    035232          XSONEF= 002000
T18XS   050542          T20DT2  066110          T38OFL  072362          T6      037726 G        XSOONL= 000100
T182SS  050642          T20EXE  064532          T38ONL  072326          T6.1    037754          XSOPED= 000010
T183SS  050706          T20EXP  064412          T38PAC  071010          T6.2    040244          XSORLL= 010000
T189EN  062022          T20EXS  064432          T38PK2  071510          T6.3    041454          XSORLS= 040000
T189FC  057346          T20LOO  062164          T38PK3  071540          T6.4    043014          XSOTMK= 100000
T189FR  061702          T20MSK  064406          T38PK4  071560          T6.5    043372          XSOVCK= 000020
T189FS  061722          T20PAC  065730          T38RES  073342          T6.6    044644          XSOWLE= 004000
T19CLE  061550          T20PK2  066100          T38SIZ  071566          T7      050040 G        XSOWLK= 000004
T19CLR  061304          T20RFI  065536          T38SSR  072206          T8      051320 G        XXCOMM  003122 G
T19CNV  061614          T20RSF  065256          T38SST  072416          T8.1    051336          X$ALWA= 000000
T19DAT  061670          T20SET  065704          T38TAD  071020          T8.2    053736          X$FALS= 000040
T19DT2  062040          T20SEX  065646          T38WLE  072065          T8.3    054624          X$OFFS= 000400
T19EXE  057532          T20SRD  065476          T38WR   071562          T8.4    056074          X$TRUE= 000020
T19EXP  057412          T20SSR  064561          T38WRL  072024          T9      062146 G        X1.COR= 020000
T19EXS  057432          T20SWP  065147          T38WRT  071740          T9.1    062164          X1.DLT= 100000
T19LOO  051336          T20WFI  065556          T39BFR  075576          UAM   = 000200 G        X1.MBZ= 017375
T19MSK  057406          T20WFM  065612          T39BSO  075570          UNITN   002202 G        X1.RBP= 000400
T19PAC  061660          T20WMI  065632          T39BS1  075571          UNREC = 000006          X1.SPA= 040000
T19PK2  062030          T20WNP  065516          T39BS2  075572          USI     004123          X1.UNC= 000002
T19PRE  057344          T202SS  064616          T39DAT  077474          WAITF   016250 G        X2.BUF= 000100
T19RFI  061416          T203SS  064662          T39DLY  075552          WC.IFA= 000200          X2.EXT= 000200
T19RSF  061350          T204SS  064727          T39DSW  076300          WC.IFE= 000002          X2.OPM= 100000
T19RST  061200          T205SS  064772          T39DTA  076270          WC.IGO= 000001          X2.RCE= 040000
T19SET  061570          T206SS  065036          T39EAI  076276          WC.IRE= 000010          X2.REV= 000077
T19SEX  061532          T208SS  065101          T39ETN  076501          WC.IRW= 000004          X2.SPA= 035400
T19SNP  061372          T23A    003144 G        T39ETS  076412          WC.IOT= 000100          X2.UNI= 000007
T19SRD  061330          T23B    003146 G        T39MCL  077414          WC.I1T= 000040          X2.WCF= 002000
T19SSR  057573          T3      025750 G        T39NBA  077035          WC.I5R= 000020          X3.DCK= 000010
```

Symbol table

```
X3.MBZ= 000006          X3.REV= 000040          X3.TRF= 000020          X4.MBZ= 017400          X4.TSM= 020000
X3.MDE= 177400          X3.RIB= 000001          X4.HSP= 100000          X4.RCE= 040000          X4.WRC= 000377
X3.OPI= 000100          X3.SPA= 000200

. ABS.  102004    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 396
Work  file writes: 361
Size of work file: 32288 Words  ( 127 Pages)
Size of core pool: 19714 Words  ( 75 Pages)
Operating  system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:12:41.72
CVTSBD,CVTSBD,=SVC/ML,CVTSBD
```